NOVEL UNCERTAINTY QUANTIFICATION TECHNIQUES FOR PROBLEMS DESCRIBED BY STOCHASTIC PARTIAL DIFFERENTIAL EQUATIONS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Peng Chen

August 2014

NOVEL UNCERTAINTY QUANTIFICATION TECHNIQUES FOR PROBLEMS DESCRIBED BY STOCHASTIC PARTIAL DIFFERENTIAL EQUATIONS

Peng Chen, Ph.D.

Cornell University 2014

Uncertainty propagation (UP) in physical systems governed by PDEs is a challenging problem. This thesis addresses the development of a number of innovative techniques that emphasize the need for high-dimensionality modeling, resolving discontinuities in the stochastic space and considering the computational expense of forward solvers. Both Bayesian and non-Bayesian approaches are considered. Applications demonstrating the developed techniques are investigated in the context of flow in porous media and reservoir engineering applications.

An adaptive locally weighted projection method (ALWPR) is firstly developed. It adaptively selects the needed runs of the forward solver (data collection) to maximize the predictive capability of the method. The methodology effectively learns the local features and accurately quantifies the uncertainty in the prediction of the statistics. It could provide predictions and confidence intervals at any query input and can deal with multi-output responses.

A probabilistic graphical model framework for uncertainty quantification is next introduced. The high dimensionality issue of the input is addressed by a local model reduction framework. Then the conditional distribution of the multi-output responses on the low dimensional representation of the input field is factorized into a product of local potential functions that are represented

non-parametrically. A nonparametric loopy belief propagation algorithm is developed for studying uncertainty quantification directly on the graph. The nonparametric nature of the model is able to efficiently capture non-Gaussian features of the response.

Finally an infinite mixture of Multi-output Gaussian Process (MGP) models is presented to effectively deal with many of the difficulties of current UQ methods. This model involves an infinite mixture of MGP's using Dirichlet process priors and is trained using Variational Bayesian Inference. The Bayesian nature of the model allows for the quantification of the uncertainties due to the limited number of simulations. The automatic detection of the mixture components by the Variational Inference algorithm is able to capture discontinuities and localized features without adhering to ad hoc constructions. Finally, correlations between the components of multi-variate responses are captured by the underlying MGP model in a natural way.

A summary of suggestions for future research in the area of uncertainty quantification field are given at the end of the thesis.

BIOGRAPHICAL SKETCH

The author was born in the city of Yangzhou, Jiangsu Province, China, in Oct, 1988. After completing his high school education from Yangzhou Middle School, the author was admitted into the department of Material Science and Engineering at Huazhong University of Science and Technology (HUST) in 2005, from where he received his Bachelor's degree in June, 2009. Then, the author entered the doctoral program at the Sibley School of Mechanical and Mechanical and Aerospace Engineering, Cornell University, and was awarded a Master's degree in January, 2013.

This thesis is dedicated to my parents Chen, Yongping and Yuan, Liangping for their constant support and encouragement towards academic pursuits during my school years.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my research advisor, Professor Nicholas Zabaras, for his constant support, guidance and patience over the past five years. He helped me build a solid background on my research, and more important, he taught me how to think critically and how to express myself efficiently. He influenced me not only on the academic life, but also on other aspects towards to a successful career in the future. I appreciate all his helps to make my Ph.D. experience invaluable, and I believe what I learned from him will become precious treasures for my future career.

I would also like to thank Professors Gennady Samorodnitsky and Derek H. Warner for serving on my special committee and for their encouragement and suggestions on the courses of this work. Their kindly helps are precious to me.

This research was supported by an OSD/AFOSR MURI09 award on uncertainty quantification, the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and the Computational Mathematics program of the National Science Foundation (NSF) (award DMS-0809062). This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Additional computing resources were provided by the NSF through TeraGrid resources provided by NCSA under grant number TG-DMS090007.

I would like to thank the Sibley School of Mechanical and Aerospace Engineering for having supported me through a teaching assistantship for part of my study at Cornell. Finally, I would like to thank fellow MPDC members and my brothers of CDSC for their support during my days at Cornell.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Uncertainty Quantification (UQ) is vital in studying physical problems in all engineering and scientific fields. UQ is a broad topic involving many aspects, for example, representation of uncertainty, propagation of uncertainty across scales, validation and verification for predictive computational science, visualization of uncertainty in high-dimensional spaces and so on [42, 22, 69, 68, 76]. The focus of this thesis is to develop efficient methodologies for investigating the propagation of uncertainty in general physical problems with stochastic inputs. These stochastic input conditions mostly arise from uncertainties in boundary and initial conditions as well as from inherent random material heterogeneities. To accurately predict the performance of physical systems, it becomes essential for one to include the effects of input uncertainties into the model system and understand how they propagate and alter the final solution.

Generally, the deterministic physical system can be described by governing equations in the form of ordinary/partial differential equations (ODEs/PDEs). The presence of uncertainties can be modeled in the system through reformulation of the governing equations as stochastic ordinary/partial differential equations (SODEs/SPDEs). The particular physical problem of interest in this thesis is the porous media flow problem in reservoir domain, which is also called reservoir simulation. This problem involves multiscale/multiphysics effects that have attracted the attention of a numerous researchers during the past few decades. Modern reservoir characterization and geostatistical modeling techniques aim at integrating information from different scales to build high resolution models with multi-million cells that describe the heterogeneous reservoir

properties in great detail. Multi-phase flow through these highly detailed reservoirs is then studied using either Finite Element (FEM) or Finite Volume (FVM) methods. However, simulation times are typically prohibitively large for uncertainty propagation or inversion tasks. To decrease the computational costs one resorts to multiscale methods [50, 70, 48]. These exploit the separation of scales and result in accelerated simulations.

Randomness in reservoir modeling is not intrinsic. That is, the permeability or porosity fields required for the construction of a reservoir are very specific physical quantities. However, we are unable to fully resolve these quantities experimentally. This is exactly why we are forced to treat them as uncertain. That is, the uncertainties involved represent a knowledge gap. The goal of uncertainty propagation is to quantify how this input uncertainty propagates (UP task) to the quantities of interest such as the pressure or the velocity response fields. There are three main difficulties associated with the representation of uncertainties in field quantities for reservoir modeling. Firstly, getting subsurface experimental measurements is very expensive. Therefore, only a limited number of observations is available for this purpose. Secondly, both field quantities are very high-dimensional in nature. This mandates the use of dimensionality reduction techniques. Hence, the problem is to quantify the fields' uncertainty based on a limited set of experimental observations with as few variables as possible. Towards this goal, we employ the well-known Karhunen-Loève expansion [62]. Thirdly, due to the heterogeneous nature of the reservoir properties, there is high probability of the existence of strong local features or discontinuities in the response surface. Such features could not be easily discovered or modeled. Therefore, in this thesis, we will try to address such difficulties for the UP problems starting from simple stochastic elliptic problems and to proceed-

ing reservoir simulation problems.

Over the past few decades, many methods and algorithms have been developed to address such UP problems. The most traditional one is the Monte Carlo (MC) method. Its wide acceptance is due to the fact that it can compute the complete statistics of the solution, while having a convergence rate that is independent of the input dimension. Nevertheless, it quickly becomes inefficient in high dimensional and computationally intensive problems, where only a few samples are available.

Another well-known approach for the UP task is the spectral finite element method [39, 39]. It involves the projection of the response on a space spanned by orthogonal polynomials of the random variables and the solution of a system of coupled deterministic equations involving the coefficients of the expansion in these polynomials. The scheme was originally developed for Gaussian random variables which correspond to Hermite polynomials (polynomial chaos (PC)). It was later generalized to include other types of random variables (generalized PC (gPC)) [117], and then expanded to the multi-element case. The multi-element generalized polynomial chaos (ME-gPC) method [109, 110] decomposes the stochastic space in disjoint elements and then employs gPC on each element. The coupled nature of the resulting equations that determine the coefficients of the polynomials make the application of the method to high input dimensions rather difficult [33].

Stochastic collocation methods extend upon the ideas of the SFEM, but do not require any intrusive changes to the simulator since they are approximating the PC coefficients by numerical integration. The response is represented as an interpolative polynomial of the system response (output) in the random

input space constructed by calls to the computer code at specific input points. In [6, 74], a Galerkin based approximation was introduced alongside a collocation scheme based on a tensor product rule using one-dimensional Gauss quadrature points. These methods, however, suffer from the curse of dimensionality, albeit they can deal with higher dimensions than SFEM. To address high dimensionality problems, various sparse grid collocation (SGC) methodologies were developed based on the Smolyak algorithm [92]. In [64], the authors developed an adaptive hierarchical sparse grid collocation algorithm and considered a number of applications with non-smooth behavior in the stochastic space. However, the piecewise local linear nature of the scheme performed poorly when only a few data points were used while interpolation of adverse functions was shown that it can trick the adaptive algorithm into stopping prior to convergence.

We have found that a local approach to uncertainty propagation is efficient to capture localized features in the stochastic space, assuming that one selects within each local model the most informative input to maximize predictive capability. In [41, 9], the authors developed such kind of method, specifically, a treed Gaussian process model where on each leaf of the tree, Bayesian Experimental Design techniques were used to learn a multi-output Gaussian process. The active learning aspects of these Bayesian approaches was shown to lead to better convergence than interpolation-based methods such as adaptive sparse grids [9].

Locally weighted projection regression (LWPR) is an algorithm for incremental nonlinear function approximation in high-dimensional spaces [21, 86, 83]. At its core, it employs nonparametric partial least squares regression to locally ap-

proximate the relationship between input and output. This methodology has several merits including no need to memorize the training data, adjusting the local models only by the local information, an ability to deal with high dimensional correlated data, and providing a confidence interval for each prediction. However, there still exist several problems that limit its application to uncertainty quantification tasks, for example, (1) the accuracy of this approach cannot be guaranteed, (2) the training data points are randomly sampled, and (3) the learning process is not optimized. Hence, in this work, we propose an adaptive way to improve the learning process of the LWPR method, in order to solve the aforementioned problems with emphasis on uncertainty quantification tasks. For brevity, we name the method as the *adaptive locally weighted projection regression* (ALWPR) method.

This new framework demonstrated that ASGC can be outperformed. Similar conclusion were obtained in [9] by a tree of Gaussian processes (tGP) and in [10] using a tree based on Relevance Vector Machines (RVM) with gPC basis functions. The local features or discontinuities could be efficiently captured under an active learning scheme. The selection of new sample input is based on the predictive variance and an additional distance penalty term. The method works for any input distribution and can provide predictions with error-bars at any query point. This framework has excellent performance in the low-dimensional stochastic elliptic problem, nevertheless, it still suffers from the curse of dimensionality, which prevented us from applying it to real flow problems. Therefore, we further construct a new UQ framework from a completely different perspective, inspired by the work in [58, 56]. The framework is widely referred to as *probabilistic graphical models*.

Probabilistic graphical models [58] provide a powerful framework that effectively interprets complex probabilistic relations between many inter-correlated variables. The two basic elements of a graphical model are its nodes and edges. The nodes represent the random variables and edges linking nodes represent correlations between them. The joint probability distribution can be accessed by decomposing the complex network into local clusters defined by connected subsets of nodes. Then, by applying appropriate inference algorithms, the marginal and conditional probabilities of interest can be effectively calculated.

Probabilistic graphical model has been used in a range of application domains, which include web search [59], medical and fault diagnosis [5], speech recognition [13], robot navigation [102], bioinformatics [7], communications [18], natural language processing [56], computer vision [54], and many more. Most of these applications involve discrete random variables or low-dimensional continuous random variables. However, for problems involving high-dimensional continuous variables, the number of efficient and accurate algorithms is limited.

The general procedure of studying a graphical model problem can be summarized as follows: (1) Study the problem and design the structure of the graphical model; (2) Select suitable model reduction techniques to reduce the dimensionality of the inputs; (3) Prepare the training data; (4) Learn the parameters of graphical model with the training data; (5) Solve an inference problem, that is, find the conditional or marginal probabilities of interest. In this framework, we consider solving a single phase flow through porous (heterogeneous) media. The stochastic input comes from the permeability field, while the responses of interest are the velocity and pressure. The goal is to construct a probabilistic

graphical model to capture the probabilistic relationship between the stochastic input to the responses.

In the designed framework, the structure of the probabilistic graphical model is derived from the FEM mesh, with the node on the FEM mesh replaced by the random variables and the deterministic relationship between nodes replaced by the correlation functions. The correlations are modeled nonparametrically correlations (non-Gaussian). All the unknown parameters of the graphical model can be learned locally via techniques such as maximum likelihood (MLE), or maximum a posterior probability (MAP). With all the parameters completely known, the proposed framework allows us to investigate the UP problem by running an inference algorithm directly on the graph. In addition, it can also act as a surrogate model to the deterministic solver, that is, for any realization of the input permeability, it can give us the predictions of physical responses as well as the confidence on these predictions (induced by the limited data used to train the graphical model). There exists a number of inference algorithms, but they can be divided into two groups, sampling based inference or variational inference. Since the designed graphical model is nonparametric, a sampling-based nonparametric belief propagation [88, 93] algorithm is employed in this work to carry out the inference task.

In [108], the authors proposed a probabilistic graphical model for multiscale stochastic partial differential equations (SPDEs) that focuses on the correlation between physical responses. The distribution of physical responses conditioned on stochastic input was approximated using conditional random field theories. Different physical responses (such as flux and pressure in flows in heterogeneous media) are correlated in such a way that their interactions are assumed to

be conditioned on fine-scale local properties. No model reduction of fine-scale properties was involved in this process. The influence of fine-scale properties on coarse-scale responses was modeled through a set of hidden variables. The approach in this thesis is significantly different in multiple fronts: (1) the graphical model considers output responses that are independent of each other; (2) an explicit model reduction scheme is considered to reduce the dimensionality of the random permeability field without the need for introducing hidden variables; and (3) the graph structure and graph learning scheme are implemented in a completely different algorithmic approach based on the Expectaction/Maximization (EM) algorithm and a sampling based approach to nonparametric belief propagation.

In this approach, a relatively high-dimensional problem can be solved while at the same time it can represent non-trivial correlations between various variables. However, we notice that this approach remains computationally expensive and hard to implement while problems are observed when applying it to problems with localized features or discontinuities. Also,the correlations between different responses could not be captured by this approach. These drawbacks will be further addressed next by a mixture of Gaussian Processes (GPs) model that would allows to address realist flow problems.

In most studies of porous media flow, only the uncertainty of the permeability field is taken into account [66, 11]. The other important quantity, the porosity of the rock, is usually assumed to be constant. This is, partly, justified because of the uncertainty in porosity is at least one order of magnitude smaller than the uncertainty in permeability [23]. In addition, including both fields practically doubles the dimensionality of the uncertainty propagation problem. To the best

of our knowledge, [40] is the first effort to simultaneously treat as random fields both permeability and porosity. In this thesis, we do the same. However, we construct a realistic model that is based on data instead of a synthetic one.

The issue of limited data is best captured by the ideas outlined in [11]. The authors use a fully Bayesian framework and are able to quantify the epistemic uncertainty induced by the limited number of simulations to the statistics of interest. The model they adopt for this purpose is the multi-output Gaussian process (MGP) of [26]. The added benefit of using MGP is that it is able to capture the linear part of the correlations between distinct outputs (e.g., pressure and the velocity field) in a natural way. The use of a separable covariance function in a random, spatial and time component makes possible the utilization of linear algebra tricks that avoid the construction and inversion of large covariance matrices.

In order to be able to capture non-stationary effects such as localized features and discontinuities, we extend the MGP model in a non-trivial way. In particular, we consider an infinite mixture of MGP's. Similar ideas have been used in various fields, e.g. L*á*zaro-Gredilla et al. [61] solve with a similar model a multi-target tracking problem, Ross and Dy [80] solve the lung disease subtype identification problem, Yuan and Neubauer [118] learn robot kinematics and Sun and Xu [95] study a traffic flow problem. The first step in a mixture model, is to assign a label to each observation indicating to which one of the components of the mixture it belongs to. This label is treated as a latent random variable. Then, observations with the same label are grouped together in a single MGP model. To allow for an arbitrary number of mixture components, we make use of a Dirichlet process (DP) prior [97, 98, 15]. The posterior of all

the parameters of the model is constructed by employing Variational Inference (VI) techniques [14, 95]. VI techniques approximate the posterior by minimizing its Kullback-Leibler divergence (information loss) from a parametrized family of candidate distributions. In our case, we derive fast approximation schemes that lead to a convergence rate that is orders of magnitude faster than traditional Markov Chain Monte Carlo (MCMC) sampling of the posterior. After approximating the posterior with VI, the UP problem can be solved by using the probabilistic surrogate. As in [11], the Bayesian nature of our model allows for the quantification of the uncertainty due to limited simulations.

The thesis is organized as follows. In Chapter 2, the adaptive locally weighted projection regression method is introduced. In Chapter 3, the probabilistic graphical model is discussed in details. In Chapter 4, an infinite mixture of Gaussian process model is constructed, and an variational inference algorithm is developed to approximate the posterior distribution. Finally, in Chapter 5, conclusions of this thesis work and suggestions for future research are summarized.

CHAPTER 2

ADAPTIVE LOCALLY WEIGHTED PROJECTION REGRESSION METHOD FOR UNCERTAINTY QUANTIFICATION

In the chapter, the mathematical framework of the stochastic problem is firstly introduced followed by a brief review of the LWPR method in Section 2.1.1. The ALWPR algorithm is described in detail in Section 2.1.2. Various examples are given in Section 2.2 demonstrating the accuracy and efficiency of the ALWPR method when applied to UQ tasks. Brief conclusions are provided in Section 2.3.

2.1 Methodology

Let us define a complete probability space $(\mathcal{X}, \mathcal{F}, \mathcal{P})$ with sample space $\mathcal{X}$ which corresponds to the outcomes of some experiments, $\mathcal{F}$ is the $\sigma-$algebra of subsets of $\mathcal{X}$ and $\mathcal{P} : \mathcal{F} \to [0, 1]$ is the probability measure. We assume that the stochastic problem has been formulated in such a way that $\mathcal{X}$ is a compact subset of $\mathbb{R}^K$ for some $K \geq 1$:

$$\mathcal{X} = \times_{k=1}^{K} [a_k, b_k], \tag{2.1}$$

with $-\infty \leq a_k < b_k \leq +\infty$ as the upper and lower bounds of each dimension. The underlying $\sigma -$ algebra is then:

$$\mathcal{F} = \{B \cap \mathcal{X} : \forall B \in \mathcal{B}^K\}, \tag{2.2}$$

where $\mathcal{B}^K$ is the Borel $\sigma -$algebra of $\mathbb{R}^K$. Then, we let $\mathcal{P}$ be absolutely continuous (with respect to the underlying Lebesgue measure), i.e. there exists a density function $p : \mathcal{X} \to \mathbb{R}$ s.t. for any $A \in \mathcal{F}$ we have

$$P(A) = \int_A p(\mathbf{x}) d\mathbf{x}. \tag{2.3}$$

Let us now consider the multi-output function $\mathbf{f} : \mathcal{X} \to \mathbb{R}^M$ representing the result of a deterministic solver modeling a physical system, i.e. at a given input point $\mathbf{x} \in \mathcal{X}$ the predicted response of the system is $\mathbf{f}(\mathbf{x})$. We will write

$$\mathbf{f} = (f_1, \ldots, f_M), \tag{2.4}$$

where f_r is the $r - th$ output of the response function, $r = 1, \ldots, M$. In this work, we assume there is no modeling error. The input distribution induces a probability distribution on the output. The UQ problem involves the calculation of the statistics of the output $\mathbf{y} = \mathbf{f}(\mathbf{x})$. Quantities of particular interest are the q-moments $m^q = (m_1^q, \ldots, m_M^q)$ for $q \geq 1$ and $r = 1, \ldots, M$:

$$m_r^q := \int_{\mathcal{X}} f_r^q(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}. \tag{2.5}$$

In particular, the mean $\mathbf{m} = (m_1, \ldots, m_M)$:

$$\mu_r := m_r^1 = \int_{\mathcal{X}} f_r(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}, \tag{2.6}$$

and the variance $\mathbf{v} = (v_1, \ldots, v_M)$:

$$v_r := \int_{\mathcal{X}} (f_r(\mathbf{x}) - \mu_r)^2 p(\mathbf{x}) d\mathbf{x} = m_r^2 - (m_r^1)^2. \tag{2.7}$$

In this work, we build a surrogate model $\tilde{\mathbf{f}}(\cdot)$ to approximate the nonlinear output function $\mathbf{f}(\cdot)$, and the aforementioned UQ problem will be investigated using the surrogate model. As it is typical with other UQ methods (e.g. sparse grids [64]), we concentrate on building a surrogate of individual responses (i.e. for each given r) without considering correlations between the output variables.

2.1.1 Local Weighted Projection Regression

The core of the LWPR [83] method is to find piecewise low-dimensional linear approximations to the nonlinear output function $\mathbf{f}(\cdot)$ (Eq. (2.4)). For the multi

output case, we assume independence of each dimension. Thus for the r-th output, we build a separate LWPR model to approximate f_r. The LWPR method combines Local Weighted Regression (LWR) and Partial Least Squares (PLS) that are briefly discussed next.

Local Weighted Regression Framework

The LWPR regression function for the $r-th$ output is constructed by blending S local linear models (so called receptive fields) $\phi_r^{(s)}(\mathbf{x})$ in the form

$$\tilde{f}_r(\mathbf{x}) = \frac{1}{W(\mathbf{x})} \sum_{s=1}^{S} w_s(\mathbf{x}) \phi_r^{(s)}(\mathbf{x}), \quad W(\mathbf{x}) = \sum_{s=1}^{S} w_s(\mathbf{x}). \tag{2.8}$$

Here, $w_s(\mathbf{x})$ is a measure of locality for each data point, which is usually modeled by a Gaussian kernel

$$w_s(\mathbf{x}) = \exp\left\{-\frac{1}{2}(\mathbf{x} - \mathbf{c}_s)^T \mathbf{D}_s (\mathbf{x} - \mathbf{c}_s)\right\}, \tag{2.9}$$

where $\mathbf{c}_s$ is the center of the s^{th} local model and $\mathbf{D}_s$ is positive semi-definite distance metric that determines the size and shape of the local model. The dependence of the weights on s is not shown here to simplify the notation.

The local linear model $\phi_r^{(s)}(\mathbf{x})$ can be built by various linear regression methods [43], such as ordinary least squares [73], principal component regression [46, 55], ridge regression [29, 96], and partial least squares [34, 84]. In general, the local linear model $\phi_r^{(s)}(\mathbf{x})$ can be expressed as:

$$\tilde{\boldsymbol{y}}_r^{(s)}(\mathbf{x}) = \boldsymbol{\phi}_r^{(s)}(\mathbf{x}) = \boldsymbol{\phi}_r^{(s)}(\mathbf{x}; \boldsymbol{\beta}_s), \tag{2.10}$$

where $\boldsymbol{\beta}_s$ are the regression associated parameters. Given a set of training data, the learning process includes the calculation of the regression parameters $\boldsymbol{\beta}_s$

and the distance metric $\mathbf{D}_s$. In this work, for mathematical convenience, we use the Cholesky decomposition of $\mathbf{D}_s$, where $\mathbf{D}_s = \mathbf{M}_s^T \mathbf{M}_s$, and learn the upper triangular matrix $\mathbf{M}_s$ instead of $\mathbf{D}_s$. With S predictions from all the local models, $\tilde{y}_r^{(s)}(\mathbf{x}_q)$, at query point $\mathbf{x}_q$, the final output of LWPR for the $r-th$ output is simply given as follows:

$$\tilde{y}_r(\mathbf{x}_q) = \frac{1}{\sum_s w_s(\mathbf{x}_q)} \sum_s w_s(\mathbf{x}_q) \tilde{y}_r^{(s)}(\mathbf{x}_q). \tag{2.11}$$

Partial Least Squares

In LWPR [83], Partial Least Squares (PLS) is chosen as the basis for the local linear models $\phi_r^{(s)}(\mathbf{x})$ (Eq. (2.10)). PLS is a regression technique that predicts the dependent response y (here taken as scalar) in terms of the K-th dimensional input $\mathbf{x}$ [34, 84, 4] (note that in this section, we work with a generic output y and all subscripts r are dropped). Let us denote with $\mathbf{y}$ $(n \times 1)$ and $\mathbf{X}$ $(n \times K)$ the centered training output and input data, respectively. PLS regression computes the directions $\mathbf{u}$ $(K \times 1)$ in the input space (also called latent vectors) that maximize the covariance between $\mathbf{X}$ and $\mathbf{y}$. The score vectors $\mathbf{z}$ $(n \times 1)$ are then formed as a linear combination of the columns of $\mathbf{X}$ with weights $\mathbf{u}$ (in some sense providing the best linear combination of the columns of $\mathbf{X}$ for predicting $\mathbf{y}$). Ordinary linear regression is then performed of $\mathbf{y}$ on the score vectors. The residuals after regressing $\mathbf{y}$ and $\mathbf{X}$ are defined such that orthogonality of the latent vectors is enforced. This ensures that the multiple regressions of $\mathbf{y}$ on the score vectors can be obtained one column at a time. The algorithm iteratively reveals more and more information about the connection between $\mathbf{y}$ and $\mathbf{X}$.

The basic algorithm is summarized below [83]:

1. Initialization: $\mathbf{X}_1 = \mathbf{X}, \mathbf{y}_1 = \mathbf{y}$.
2. For $i = 1$ to R (number of latent variables) do
   (a) Find the direction that maximizes the correlations between $\mathbf{X}_i$ and $\mathbf{y}_i$: $\mathbf{u}_i = \mathbf{X}_i^T \mathbf{y}_i$.
   (b) Compute the latent variables (also called scores): $\mathbf{z}_i = \mathbf{X}_i \mathbf{u}_i$.
   (c) Compute the regression coefficient: $\beta_i = \mathbf{z}_i^T \mathbf{y}_i / (\mathbf{z}_i^T \mathbf{z}_i)$.
   (d) Update the residual after regressing $\mathbf{y}_i$ on $\mathbf{z}_i$: $\mathbf{y}_{i+1} = \mathbf{y}_i - \beta_i \mathbf{z}_i$.
   (e) Update the residual after regressing $\mathbf{X}_i$ on the score vector $\mathbf{z}_i$: $\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{z}_i \mathbf{p}_i^T$, where $\mathbf{p}_i = \mathbf{X}_i^T \mathbf{z}_i / (\mathbf{z}_i^T \mathbf{z}_i)$ (transpose of the vector of regression coefficients obtained from linear regression of the columns of $\mathbf{X}_i$ on $\mathbf{z}_i$). This step enforces the orthogonality condition $\mathbf{X}_{i+1} \mathbf{u}_i = 0$.

R is often automatically determined by tracking the mean square error of the prediction [83]. The prediction for a new input $\mathbf{x}_{new}$ is performed by essentially retracing the steps of the algorithm above. Let $\bar{\mathbf{x}}$ and $\bar{y}$ be the mean of the inputs and output. The prediction process goes through the following steps:

1. Initialization: $\mathbf{x}_1 = \mathbf{x}_{new} - \bar{\mathbf{x}}$.
2. For $i = 1$ to R do
   (a) Compute the latent variables: $z_i = \mathbf{x}_i \mathbf{u}_i$.
   (b) Update the residual of $\mathbf{x}$: $\mathbf{x}_{i+1} = \mathbf{x}_i - z_i \mathbf{p}_i^T$.

The predicted output of the local model is then formed by a linear combination of the latent variables as

$$\phi_s(\mathbf{x}_{new}) = \bar{y} + \sum_{i=1}^{R} \beta_i z_i. \tag{2.12}$$

Updating the Distance Metric of Each Receptive Field

The distance metric $\mathbf{D}_s$, controls the shape and size for the local model $\phi_r^{(s)}(\mathbf{x})$ (Eq. (2.10)). In LWPR [83], the distance metric of each local model starts from a predefined value, and then can be adjusted according to the observed data. Specifically, the distance metric for each local model can be learned individually by stochastic gradient descent using a penalized cross-validation cost function [86]:

$$J_s = \frac{1}{\sum_{i=1}^{n} w_s(\mathbf{x}_i)} \sum_{i=1}^{n} \frac{w_s(\mathbf{x}_i)(y_r(\mathbf{x}_i) - \tilde{y}_r^{(s)}(\mathbf{x}_i))^2}{(1 - w_s(\mathbf{x}_i)\mathbf{x}_i^T \mathbf{P} \mathbf{x}_i)^2} + \frac{\lambda}{K} \sum_{i,j=1}^{K} (\mathbf{D}_s)_{ij}^2, \tag{2.13}$$

where n denotes the number of data points in the training set and K denotes the dimension of the distance metric $\mathbf{D}_s$ (same as the dimension of the input), $w_s(\mathbf{x}_i)$ is defined in Eq. (2.9), $\mathbf{P}$ corresponds to the inverted weighted covariance matrix of the input data (defined in [83]), $\tilde{y}_r^{(s)}(\mathbf{x}_i)$ is the local prediction given by the s^{th} local model for the $r - th$ output, and λ is the trade-off parameter that determines the strength of the penalty term. The first term of the cost function J_s is the mean leave-one-out cross-validation error of the local model which ensures proper generalization [86]. The second regularization term penalizes the sum of squared coefficients of the distance metric $\mathbf{D}_s$ to allow smoother local predictions for the model [86]. Some details on the derivation of Eq. (2.13) are provided in Appendix A.1. Based on the cost function, the distance metric can be learned via:

$$\mathbf{M}_s^{n+1} = \mathbf{M}_s^n - \alpha \frac{\partial J_s}{\partial \mathbf{M}_s}, \tag{2.14}$$

where $\mathbf{D}_s = \mathbf{M}_s^T \mathbf{M}_s$, and $\mathbf{M}_s$ is an upper triangular matrix, α is the step size [86, 83]. The gradient of J_s can be computed analytically in terms of several sufficient statistics.

Note that the adjustment of the distance metric described above is not capable of modeling alone local features. In Section 2.1.2, we address how to adaptively define the initial D_s and subsequently control its size.

Adding a Receptive Field

In LWPR [83], if a training sample $(\mathbf{x}, y)$ does not activate any of the existing receptive fields by more than a threshold w_{gen}, i.e., $\max_s w_s(\mathbf{x}) < w_{\text{gen}}$, then a new receptive field is created (w_{gen} is defined by the user, here $w_{\text{gen}} = 0.2$). The center of the new receptive field is taken as $\mathbf{c} = \mathbf{x}$ and the initial distance metric $\mathbf{D}$ is set to a default value, $\mathbf{D}_{\text{def}}$ (usually, $\mathbf{D}_{\text{def}}$ is a diagonal matrix, as $\mathbf{D}_{\text{def}} = a\mathbf{I}_{K\times K}$, where a is problem dependent). $\mathbf{D}_{\text{def}}$ determines the initial size and shape for the local model. It can be understood as the inverse of the covariance matrix of the Gaussian kernel, as shown in Eq. (2.9).

All other regression associated parameters (β, the sufficient statistics to calculate β, and $\mathbf{P}$ (Eq. (2.13))) are initialized to predefined values (zero except the matrix $\mathbf{P}$). A suitable initialization of $\mathbf{P}$ is a diagonal matrix with $\mathbf{P}_{ii} = 1/r_i^2$, where the parameters r_i take very small value, e.g., 0.001 [83].

This approach has been shown to be robust but not very accurate. As shown in [83], although the mean squares error (MSE) for all examples considered eventually converges, it actually converges to a rather large value. Furthermore, a large amount of training data were often required to achieve convergence. In

addition, it is not appropriate to set the distance metric of each new receptive field to a default value $\mathbf{D}_{\text{def}}$. For example, in problems with strong local features, the default size of the local model is much larger than the span of the local features. One needs to select the initial distance metric based on the local environment. Given a set of training data, we discuss in Section 2.1.2 when we need to add a local model and how to select the initial distance metric.

Computing Confidence Intervals

LWPR has the ability to give us a confidence interval for the prediction at a query point [83]. The prediction for a query point $\mathbf{x}_q$ is taken as a noisy observation of the true response, where the noise comes from two sources. The first noise source models the predictive error of local models, in this work, the error bar given by the local PLS method. The second noise process accounts for the difference between predictions of local models. This term comes into the picture because of the Local Weighted Regression framework, since the final prediction is obtained by averaging all the local predictions. The overall predictive variance can be approximated as (see Appendix A.2):

$$\sigma^2_{\text{pred}} = \frac{\sum_s w_s(\mathbf{x}_q)\sigma^2_{pred,s}}{(\sum_s w_s(\mathbf{x}_q))^2} + \frac{\sigma^2}{\sum_s w_s(\mathbf{x}_q)}. \tag{2.15}$$

The first term on the righthand of the equation accumulates the predicted variances for all the local models, where $\sigma^2_{pred,s}$ is the local predicted variance for the s^{th} model [83]. The second term calculates the predicted variance due to the overlap of local models, here σ^2 is defined by

$$\sigma^2 = \frac{\sum_s w_s(\mathbf{x}_q)(\tilde{y}(\mathbf{x}_q) - \tilde{y}^{(s)}(\mathbf{x}_q))^2}{\sum_s w_s(\mathbf{x}_q)}. \tag{2.16}$$

Substituting this into Eq. (2.15) results in the following:

$$\sigma^2_{\text{pred}} = \frac{1}{(\sum_s w_s(\mathbf{x}_q))^2} \sum_s^S w_s(\mathbf{x}_q)[(\tilde{y}(\mathbf{x}_q) - \tilde{y}^{(s)}(\mathbf{x}_q))^2 + \sigma^2_{pred,s}]. \tag{2.17}$$

2.1.2 Adaptive LWPR

Given a set of training data, one can build the LWPR model as summarized in Algorithm 1. w_{gen} is the main parameter of this algorithm controlling when a sample point $\mathbf{x}_i$ is to be sent to a particular local model s via the criterion $w_s(\mathbf{x}_i) > w_{gen}$. In this paper, our interest is on developing an adaptive LWPR algorithm (Algorithm 2). Based on the standard LWPR model, the following question needs to be addressed: If we are to choose the next observation, what is the most informative input we should select from the input distribution? This is the classical experimental design or active learning problem [36, 72]. In [60], the authors concluded that the most informative data point is the one that maximizes the error bar (predictive variance). In this work, we adaptively select the sample which has the maximum predictive variance. However, if the response surface has a local feature (e.g. a discontinuity), then the largest predictive variance always occurs around the local feature. This results in a cluttering of points in the training set around the local features with insufficient observations at other locations. This situation can be avoided by adding a distance penalization factor $\eta(\mathbf{x}_i)$ [67] (Eq. (2.18)) to prevent samples from lying too close to the current training data set. The scaling γ in the definition of the penalization factor attempts to balance the goal of sampling in areas of large predictive variance with the ability to detect unexplored (less-sampled) areas.

The penalty function $\eta(\mathbf{x}_i)$ is introduced with the following properties (Al-

Algorithm 1: The complete LWPR algorithm

Initialize LWPR model with no local models

for $i = 1, \ldots, l_0$ **do**

  for $s = 1, \ldots, S_{curr}$ (S_{curr} is the number of current local models) **do**

    Calculate the weight $w_s(\mathbf{x}_i)$ from Eq. (2.9)

    if $w_s(\mathbf{x}_i) > w_{gen}$ **then**

      Update the PLS regression parameters to include $\mathbf{x}_i$ (Section 2.1.1)

      Update the distance metric $\mathbf{D}_s$ to include $\mathbf{x}_i$ (Section 2.1.1)

    end if

  end for

  if no current local model was activated by more than w_{gen} **then**

    Create a new local model centered at $\mathbf{x}_i$ with an initial distance metric $\mathbf{D}_{\text{def}}$ (Section 2.1.1)

  end if

end for

gorithm 2). At first, it should give a very small value when there exists a data point in the training set that is very close to the candidate sample $\mathbf{x}_i$. In that way, this candidate point will not be selected simply because the predictive variance at this point is large. Secondly, $\eta(\mathbf{x}_i)$ should take a relatively large value if no points in the training set are close to the candidate sample. In this work, the penalty function contains a parameter γ that controls its strength. After several tests, we here select $\gamma = 10^{-2}$. We can now select as a candidate sample for our regression scheme the sample h out of N samples from the distribution $p(\mathbf{x})$ that maximizes the weighted predictive variance (Eq. (2.19)). Therefore, one can build an adaptive version of LWPR (ALWPR) by selecting the most informative

input samples from the input space $\mathcal{X}$ (Eq. (2.1)) biased by the input probability distribution.

The convergence criterion $\xi < \delta$ (with δ a given tolerance that defines a stopping criterion for the Algorithm 2) is chosen with ξ defined in Eq. (8) as the average weighted predictive variance $\sigma_i^2 \eta_i$ over the whole domain biased by the input distribution. As we know, the predictive variance around local features is higher than in other regions, however, the probability that candidate samples go to the local feature region is relatively small (discontinuities have zero probability measure). As the number of data points observed increases, ξ will gradually decrease leading to higher reliability of the prediction over the whole domain.

The Algorithm 2 starts with l_0 training points and constructs an initial LWPR model (see Algorithm 1). The value of l_0 is selected based on the input dimensionality (e.g., in this work, we set $l_0 = 50$ for $K = 2$, and $l_0 = 100$ for $K = 4$). Then N candidate points (here $N = 1000$) are sampled from the input distribution $p(\mathbf{x})$. From them, we add to the training data set the input point $\mathbf{x}_{new}$ that maximizes the weighted predictive variance (Eq. (2.19)). The next algorithmic step is updating the LWPR model using the new training set $\{\mathbf{x}_{new}, y_{new}\}$. This step includes the creation of a new receptive field if $\max_s w_s(\mathbf{x}_{new}) < w_{gen}$, or otherwise updating all neighboring local models. These calculations were discussed in the earlier sections.

If a new receptive field is created at a point at $\mathbf{x}_{new}$, the prediction error at that point will be close to zero (not exactly zero owing to the contribution from other local models). Let us assume now that no new receptive field was created at $\mathbf{x}_{new}$. In this case, we will need to check if the update of neighboring local models

Algorithm 2: The complete Adaptive LWPR framework

Start with l_0 initial training points, construct a LWPR model (Algorithm 1).

while $\xi > \delta$ **do**

Randomly sample N data points from $p(\mathbf{x})$.

Calculate the predictive variance $\sigma^2(\mathbf{x}_i)$ for each sample $\mathbf{x}_i$.

Calculate the distance factor $\eta(\mathbf{x}_i)$ as

$$\eta(\mathbf{x}_i) = 1 - \exp\left\{-\gamma \min_j (\mathbf{x}_i - \mathbf{x}^{(j)})^T (\mathbf{x}_i - \mathbf{x}^{(j)})\right\}, \tag{2.18}$$

where $\mathbf{x}^{(j)}$ is one point in the training set.

The one that maximizes the weighted predictive variance is chosen,

$$\mathbf{x}_{new} = \mathbf{x}_h, \text{ where } h = \arg\max_i (\sigma^2(\mathbf{x}_i) p(\mathbf{x}_i) \eta(\mathbf{x}_i)), \tag{2.19}$$

Calculate $\xi = \int \sigma^2(\mathbf{x}) \eta(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \approx \frac{1}{N} \sum_{i=1}^{N} \sigma^2(\mathbf{x}_i) \eta(\mathbf{x}_i) p(\mathbf{x}_i)$.

Update the LWPR model with the new training data $\{\mathbf{x}_{new}, y_{new}\}$.

if $|\widetilde{y}_{new} - y_{new}| > \epsilon$ **then**

for $s = 1, ..., S_{curr}$ **do**

Calculate the weights $w_s(\mathbf{x}_{new}) = \exp\{-\frac{1}{2}(\mathbf{x}_{new} - \mathbf{c}_s)^T \mathbf{D}_s (\mathbf{x}_{new} - \mathbf{c}_s)\}$.

if $w_s(\mathbf{x}_{new}) > w_e$ **then**

Relearn the local model s and reset the distance metric as:

$$\mathbf{D}_s = \frac{1}{\alpha_s^2} I_{K \times K}, \text{ where } \alpha_s = \frac{1}{2} \sqrt{(\mathbf{x}_{new} - \mathbf{c}_s)^T (\mathbf{x}_{new} - \mathbf{c}_s)}. \tag{2.20}$$

end if

end for

Create a new receptive field centered at $\mathbf{x}_{new}$

end if

end while

provides a reasonable prediction y_{new} at $\mathbf{x}_{new}$. The parameter ϵ is introduced to control this error. In particular, when $|\widetilde{y}_{new} - y_{new}| > \epsilon$ (with $\widetilde{y}_{new}$ the estimate at $\mathbf{x}_{new}$ before the update), a new receptive field will be introduced centered at $\mathbf{x}_{new}$ with an appropriate distance metric. A user-defined parameter w_e is introduced to control the update of the neighboring local models. In particular, the models s for which $w_s(\mathbf{x}_{new}) > w_e$ are updated. The parameter w_e controls in certain way the overlap of these neighboring to $\mathbf{x}_{new}$ local models. The updated distance metric for these models s is taken as in Eq. (2.20) and is such that the influence of local model$s$ at the point $\mathbf{x}_{new}$ is decreased, i.e., the weight $w_s(\mathbf{x}_{new})$ reduces from some large value to 0.135. This value is obtained from the definition of the weight in Eq. (2.9) using the distance metric given in Eq. (2.20).

Now suppose $w_e < w_s(\mathbf{x}_{new}) < 0.135$. This means that the current model s needs to be updated. Thus the weight $w_s(\mathbf{x}_{new})$ will increase to 0.135. With a fixed center of the local model s and fixed $\mathbf{x}_{new}$, the only way for this to happen is by increasing the size of the local model s, which of course is not desirable. This implies that we need to choose $w_e > 0.135$. For w_e close to 1, only the local models that are very close to $\mathbf{x}_{new}$ will be updated. From numerical experimentation, we found that the optimal choice for all the examples reported in this paper is $w_e = 0.3$.

Yet another consideration is relearning of the neighboring local models s after the update of the new distance metric. This is because several points that were previously included in model s may not satisfy the condition $w_s(\mathbf{x}) > w_{gen}$. Thus we need to remove these points from model s (i.e. clean the sufficient statistics of local model s) and recalculate its regression parameters. Lastly, note that in order to allow reasonable local predictions, at least $K + 1$ points need to

be observed at each local model.

Depending on the input distribution $p(\mathbf{x})$, the ALWPR algorithm will result in the creation of new models for many of the added points in the non-smooth regions of the stochastic space. In general for smooth stochastic responses, this will not be the case and the ALWPR algorithm will preferably refine the parameters in current local models than adding new local models.

For the multi-output case, we assume that the output dimensions are independent from each other. Thus we construct M independent K-inputs-to-single-output ALWPR models. All ALWPR models share the same training input data. When calculating the predictive variance for a sample input $\mathbf{x}_i$, we accumulate the predictive variances of all the outputs, $\sigma^2(\mathbf{x}_i) = \sum_{r=1}^{M} \sigma_r^2(\mathbf{x}_i)$, where $\sigma_r^2(\mathbf{x}_i)$ is the predictive variance at input $\mathbf{x}_i$ for the r^{th} output. The convergence criterion in Eq. (8) is then adjusted with ξ defined as follows:

$$\xi = \int \sigma^2(\mathbf{x})\eta(\mathbf{x})p(\mathbf{x})d\mathbf{x} \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{r=1}^{M} \sigma_r^2(\mathbf{x}_i)\eta(\mathbf{x}_i)p(\mathbf{x}_i). \tag{2.21}$$

To accelerate the data selection process, instead of taking one input sample with the largest weighted predictive variance, we can take the first n input samples with the highest weighted predictive variance. Using these input points, we can run the deterministic solver independently using different processors. After all the calculations are completed, we gather all the outputs, and include these new observations into the training set to update the model. Note that for the M multi-output case, all the M LWPR models share the same input training data, but they do not have the same local models.

For $q+n+1$ processors available, we take one processor (P_0) as the root node, q processors (P_1 to P_q) to build the LWPR model for the M-output problem, and

n processors (P_{q+1} to P_{q+n}) to run the deterministic solver. At first, the root node sends the training data set to processors P_1 to P_q. After the model has been built, the root node will receive a signal from P_1 to P_q, then it will start to sample N points from the input distribution $p(\mathbf{x})$. These samples are going to be sent to P_1 to P_q to calculate the predictive variance for each output, while the root node is calculating the value of the distance penalty function for each sample. After all the predictive variances for all the outputs have been calculated and sent back to the root node, the root node sums them up, finds the first n largest weighted predictive variances, and calculates ξ using Eq. (2.21). Finally, the new selected n inputs will be sent to processors P_{q+1} to P_{q+n} to run the deterministic solver. For each output, we check the prediction for the newly added points. If it is not satisfying the set accuracy requirements at a newly added point, a local model centered at this point will be added and the size of the neighboring local models will manually be changed. The process is repeated until convergence. The calculated responses are sent to the root node and added to the training set.

2.2 Numerical Examples

The examples considered here are designed to demonstrate that ALWPR has the ability to learn local features in the stochastic space such as discontinuities, adaptively choose the inputs for the model (active learning/experimental design), and accurately provide predictions with uncertainty for a new input.

All the examples are run on massively parallel computers at the National Energy Scientific Computing Center (NERSCC) [37].

2.2.1 Kraichnan-Orszag (K-O) problem

The transformed Kraichnan-Orszag three-mode problem is expressed as the following dynamical system [109, 64]

$$\begin{aligned} \frac{dy_1}{dt} &= y_1 y_3, \\ \frac{dy_2}{dt} &= -y_2 y_3, \\ \frac{dy_3}{dt} &= -y_1^2 + y_2^2, \end{aligned} \tag{2.22}$$

subject to initial conditions

$$y_1(0) = Y_1(0;\omega),\ y_2(0) = Y_2(0;\omega),\ y_3(0) = Y_3(0;\omega). \tag{2.23}$$

The problem exhibits a bifurcation on the parameters $y_1(0)$ and $y_2(0)$, in particular a discontinuity occurs when the initial conditions cross the planes of $y_1 = 0$ and $y_2 = 0$. The deterministic solver we use is a 4-th order Runge-Kutta method as implemented in the GNU Scientific Library [37]. We solve the system for the time interval $[0, 10]$ and record the responses at time step interval of $\Delta t = 0.01$. This results in a total of $M = 300$ outputs (100 for each of the three dimensions of the response). The error of the statistics will be evaluated using the (normalized) L_2 norm of the error in variance defined by:

$$E_{L_2} = \frac{1}{M}\sum_{r=1}^{M}(v_{r,MC} - \widetilde{v}_r)^2, \tag{2.24}$$

where $v_{r,MC}$ is the Monte Carlo estimate of the variance using 10^6 samples, and $\widetilde{v}_r$ is the predictive variance, $r = 1, \ldots, M$.

For brevity, we only consider here the two dimensional case. The initial conditions for the problem are taken as:

$$y_1 = 0,$$

$$y_2 = 0.1x_1,$$

$$y_3 = x_2,$$

where

$$x_i \sim \mathcal{U}([-1, 1]),\ i = 1, 2. \tag{2.25}$$

This problem has a line discontinuity at $x_1 = 0$. The algorithm starts with 50 random samples. Figs. 2.1-2.3 show the comparison of the prediction of $y_3(t = 10)$ with the true response at tolerance levels $\delta = 10^{-5}, 10^{-7}$ and 10^{-9}, respectively. As shown in these figures, with decreasing δ, most of the new samples selected by the algorithm are placed near the discontinuity that is gradually being resolved. Fig. 2.4 depicts the mean weighted predictive variance and L_2 norm of the error in variance as a function of the number of observations. Here, we also compare the results obtained from the ALWPR with those of the Adaptive Sparse Grid Collocation Method (ASGC) [64] and Monte Carlo method. As shown in the figure, for this KO-2 problem, for the same number of samples, ALWPR leads to higher accuracy than ASGC. Fig. 2.5 plots the predictive mean and variance of $y_3(t)$ as a function of time t and compares it with the MC prediction. Finally, using 10^5 samples, Fig. 2.6 provides the kernel density estimation of the PDF of $y_2(t = 10)$ and $y_3(t = 10)$. This example is run in parallel with 24 processors ($q = 20$ and $n = 4$).

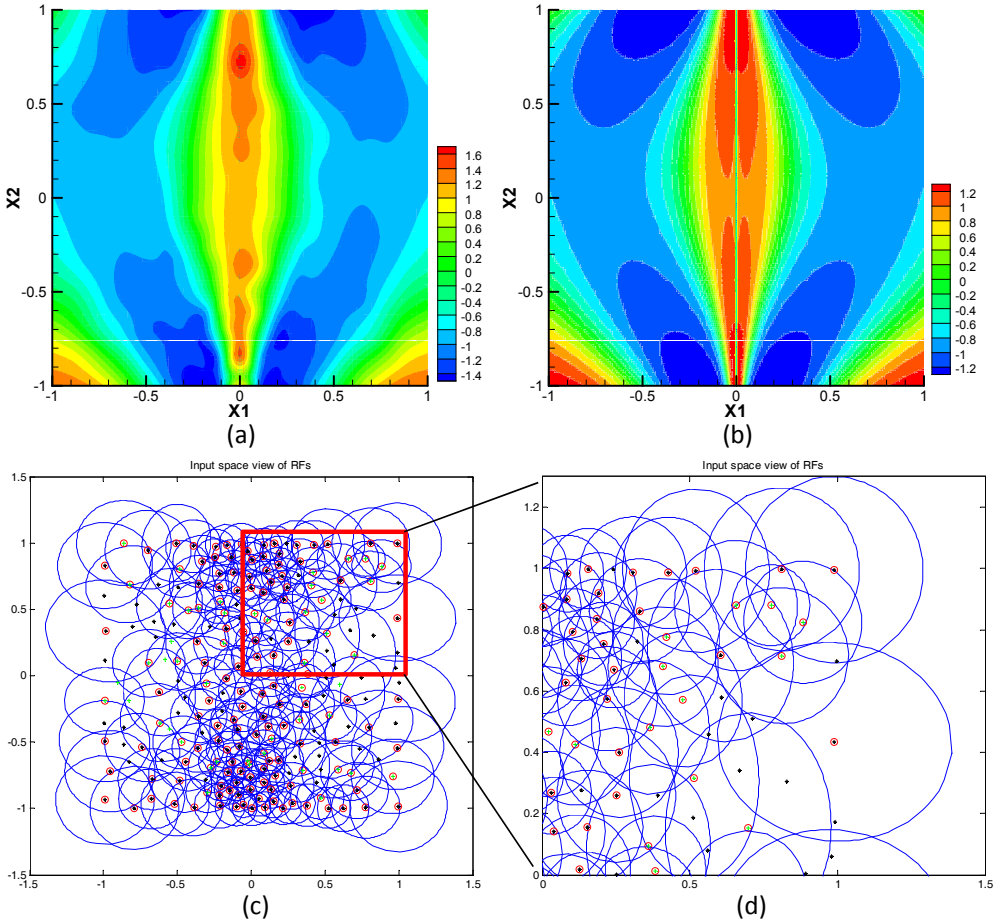

Figure 2.1: KO-2: (a) The prediction of $y_3(t = 10)$ at $\delta = 10^{-5}$. (b) The true response of $y_3(t = 10)$. (c) Final Receptive fields. (d) Initial data (red squares) and new samples selected by ALWPR (green squares).

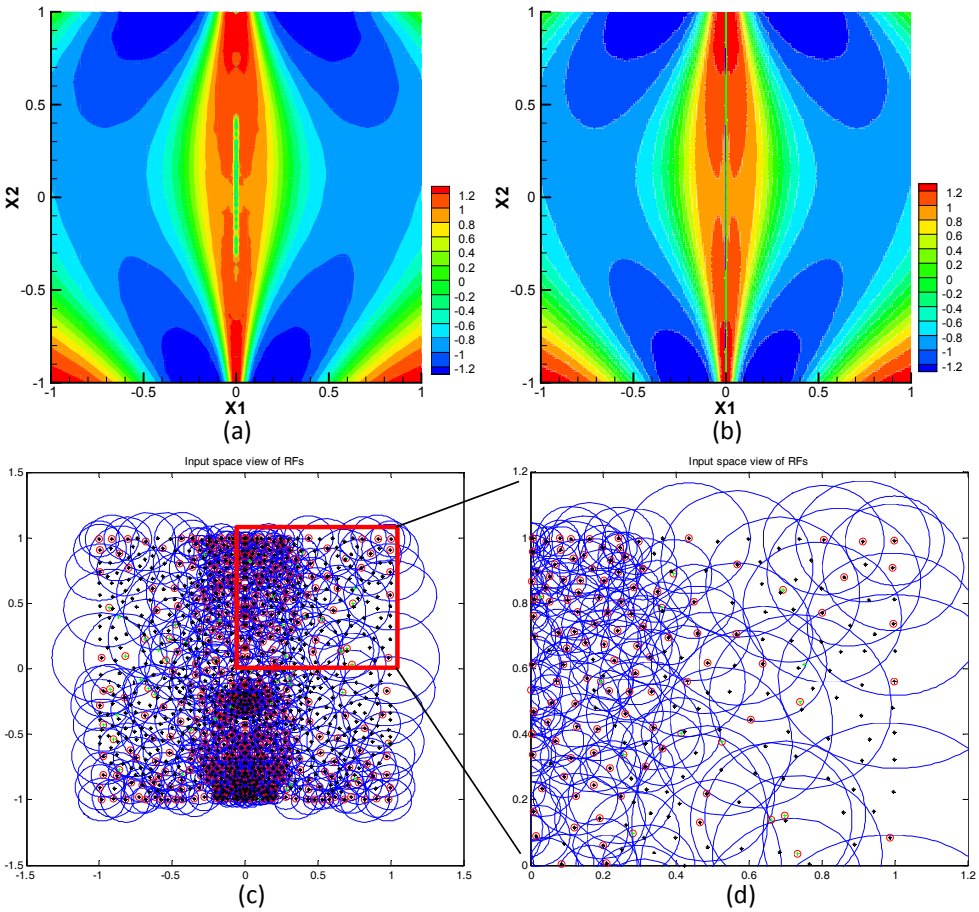

Figure 2.2: KO-2: (a) The prediction of $y_3(t = 10)$ at $\delta = 10^{-7}$. (b) The true response of $y_3(t = 10)$. (c) Final Receptive fields. (d) Initial data (red squares) and new samples selected by ALWPR (green squares).

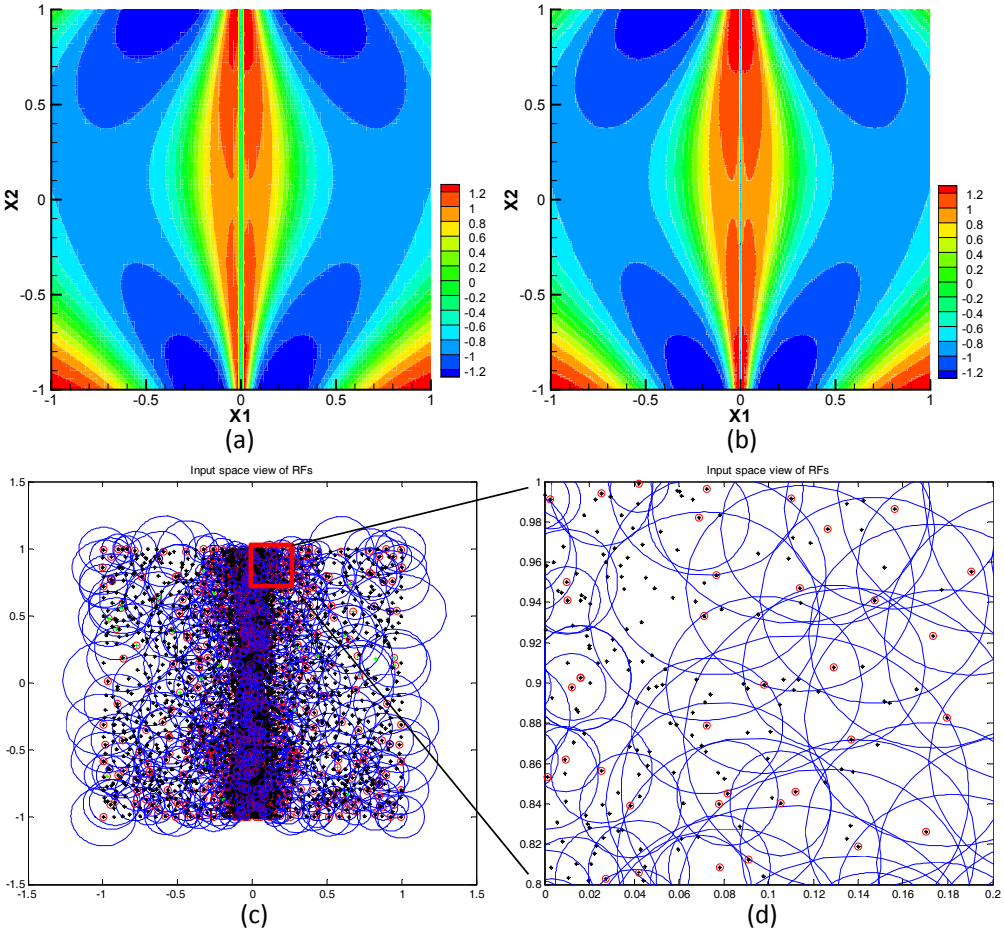

Figure 2.3: KO-2: (a) The prediction of $y_3(t = 10)$ at $\delta = 10^{-9}$. (b) The true response of $y_3(t = 10)$. (c) Final Receptive fields. (d) Initial data (red squares) and new samples selected by ALWPR (green squares).

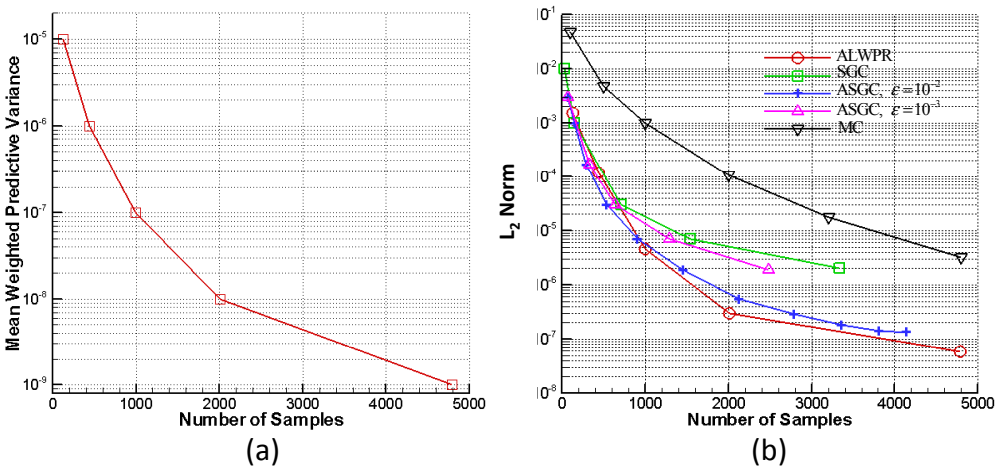

Figure 2.4: KO-2: (a) The mean weighted predictive variance as a function of the number of samples observed. (b) The L_2 norm of the error in variance as a function of the number of samples observed for ALWPR, Sparse Grid Collocation (SGC), Adaptive Sparse Grid Collocation (ASGC), and Monte Carlo (MC).

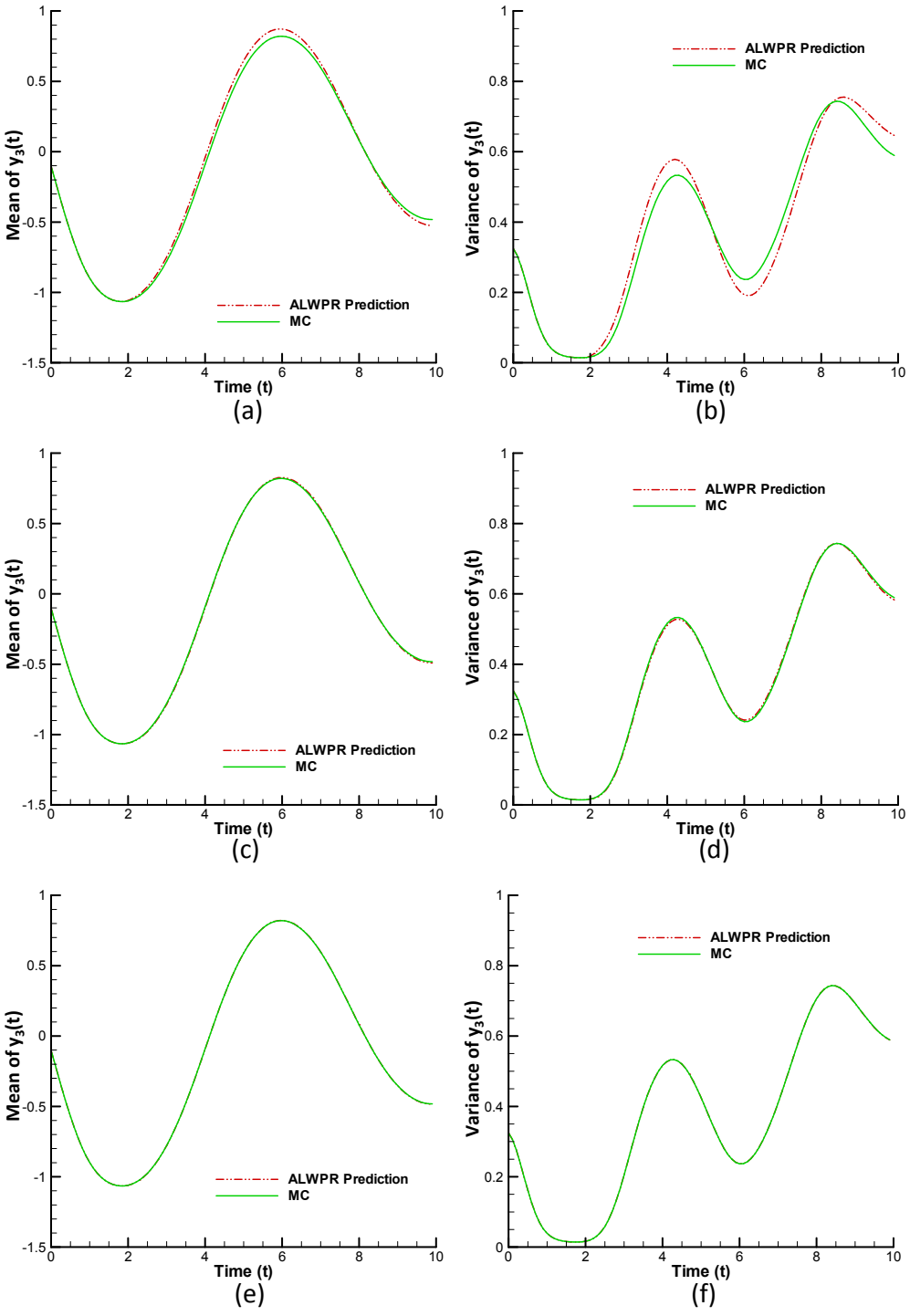

Figure 2.5: KO-2: Predictive mean (red) versus MC estimate (green) of the mean (left column) and variance (right column) of $y_3(t)$ for $\delta = 10^{-5},\ 10^{-7}$ and 10^{-9} (from top to bottom, respectively).

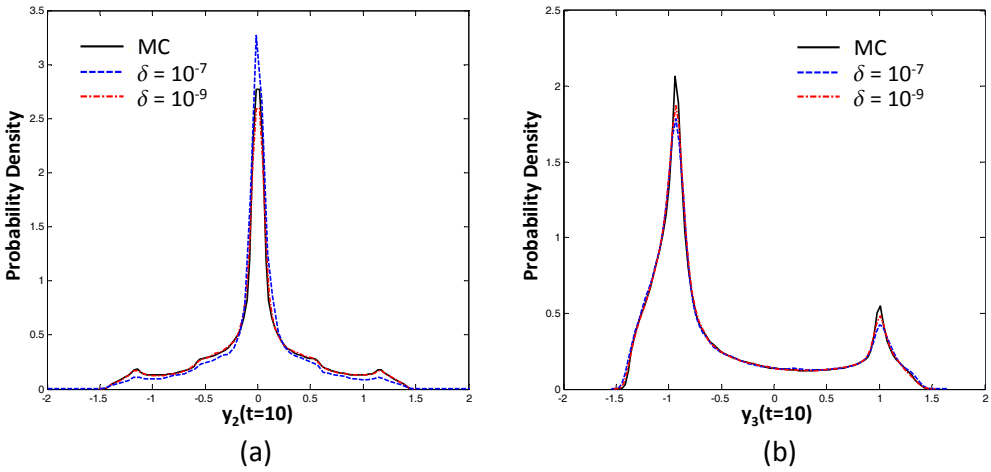

Figure 2.6: KO-2: Kernel density estimation of the PDF of $y_2(t = 10)$ (left) and $y_3(t = 10)$ (right) using 10^5 samples.

2.2.2 Horn Problem

In this section, we apply the ALWPR method to the planar acoustic horn problem [100, 105, 101], in the form of the two-dimensional Helmholtz equation in random media. The structure of the horn is depicted in Fig. 2.7. The incoming wave comes from the left end, and propagates to the right end through a horn-like tunnel. The walls of the tunnel are built by sound-hard material. The governing equations for the (complex) pressure are:

$$\nabla^2 p(x, y, \omega) + k^2(1 + n^2(x, y, \omega))p(x, y, \omega) = 0, \tag{2.26}$$

with boundary conditions

$$\begin{aligned} \frac{\partial p}{\partial \vec{n}} - ikp = 0, \quad & \text{on } \Gamma_1, \\ \frac{\partial p}{\partial \vec{n}} = 0, \quad & \text{on } \Gamma_2, \\ p(x, y, \omega) = f(x, y), \quad & \text{on } \Gamma_3, \end{aligned}$$

where $\vec{n}$ is the unit outer-pointing normal of the boundary, k is the wave number and $n^2(x, y, \omega)$ is the random reflectivity of the media. Γ_1 is the outer boundary, Γ_2 is the boundary of the tunnel (horn), and Γ_3 is the source boundary for the incoming wave (inlet of the horn). In this example, the random reflectivity of the media is chosen to be [116]

$$n^2(x, y, \omega) = \sum_{i=1}^{4} \xi_i(\omega)\psi_i(x, y), \tag{2.27}$$

where $\{\xi_i(\omega)\}$ are i.i.d. uniformly distributed random variables in $[0, 1]$ and the functions $\{\psi_i(x, y)\}$ are given by

$$\begin{aligned} \psi_1(x, y) &= \sin^2(2\pi x)\sin^2(2\pi y), \\ \psi_2(x, y) &= \sin^2(4\pi x)\sin^2(4\pi y), \end{aligned}$$

$$\psi_3(x, y) = \sin^2(6\pi x)\sin^2(4\pi y),$$

$$\psi_4(x, y) = \sin^2(6\pi x)\sin^2(6\pi y).$$

The deterministic problem is solved by using the FreeFEM++ software [47] with $f(x) = 1$ and $k = 0.7$. We consider a circular domain discretized with triangular elements with totally 3942 nodes, as shown in Fig. 2.8. In the context of the ALWPR method, this is a regression problem with 3942 outputs. In this example, the geometric parameters (see Fig. 2.8) are chosen as follows: $a = 1.6$, $b = 4$, $l = 4$, $d = 4$, $R = 9.6$ (the radius of the circular domain).

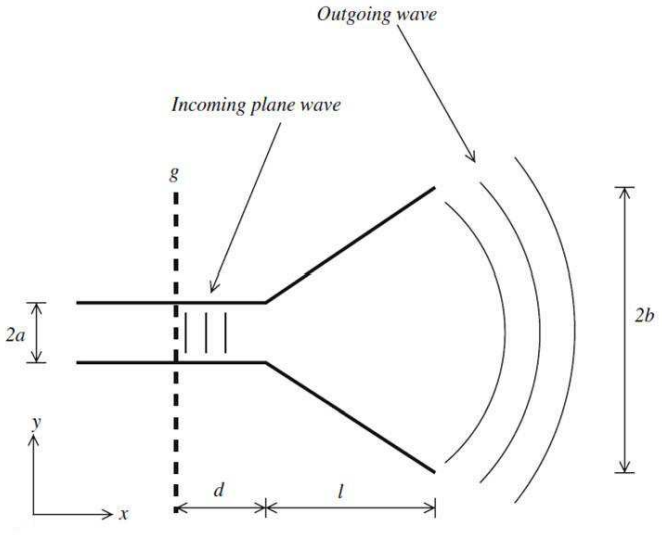

Figure 2.7: The structure of the horn, the incoming wave comes from the left end, and propagates to the right end through a horn-like tunnel, where the walls of the tunnel are built by sound-hard material.

The horn problem is studied with a four-dimensional random input using 100 initial samples. The convergence plots of the mean weighted predictive variance and L_2 norm of the error in variance are shown in Fig. 2.9. The obtained results are compared with the results of ASGC and MC method. Fig. 2.10 compares the predictive mean and variance given by ALWPR with the corre-

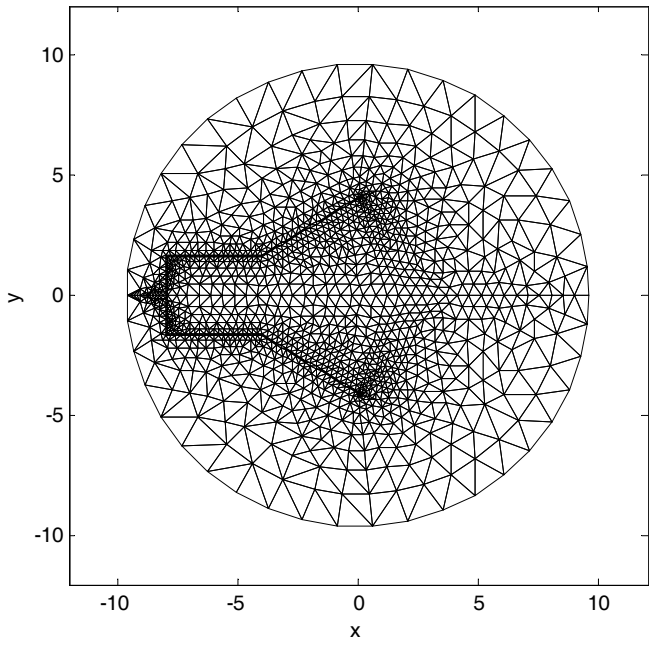

Figure 2.8: The FEM mesh used in the horn problem (3942 nodes).

sponding MC estimates obtained with 10^6 samples. Notice that as the threshold δ decreases, the predictive variance is almost identical to the MC estimates. In order to see the predictive capabilities of ALWPR for this problem, we plot the prediction at $\delta = 10^{-9}$ on two random input samples, and compare them with the true responses, as shown in Figs. 2.11 and 2.12. One can notice that the predictions agree very well with the true responses. Also to better examine the performance of ALWPR, we compare the predictive PDFs for the outputs at two specific nodes, $p(-4, 1.6)$, the junction point of the throat and horn flare, where the incoming wave first enters the divergent region of the horn, and $p(0, 4)$, the end of the horn flare where the wave leaves the horn. Fig. 2.13 provides the kernel density estimation of the PDFs at these two points by using 10^5 samples. We can see that the predicted PDFs agree well with the MC estimates. This example is run in parallel with 205 processors ($q = 200$ and $n = 5$).

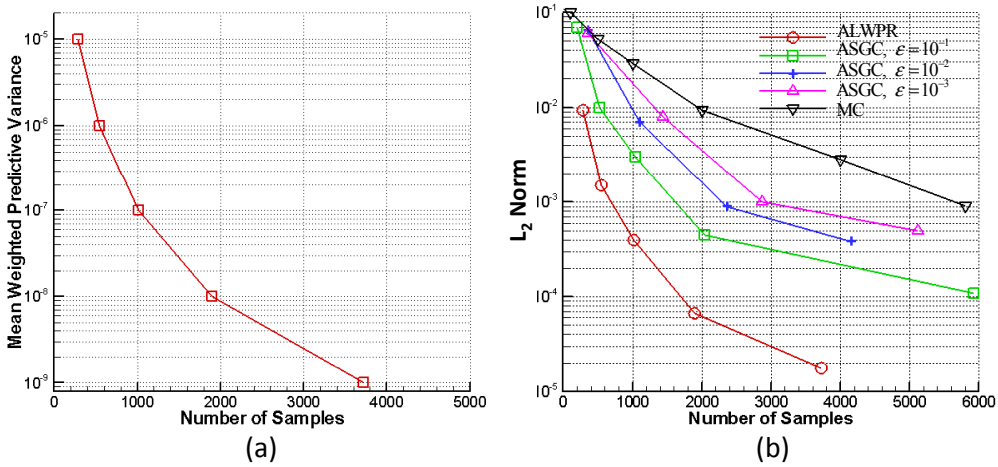

Figure 2.9: Horn (4 input dimensions): (a) The mean weighted predictive variance as a function of the number of samples observed. (b) The L_2 norm of the error in variance as a function of the number of samples used by ALWPR, ASGC and MC.

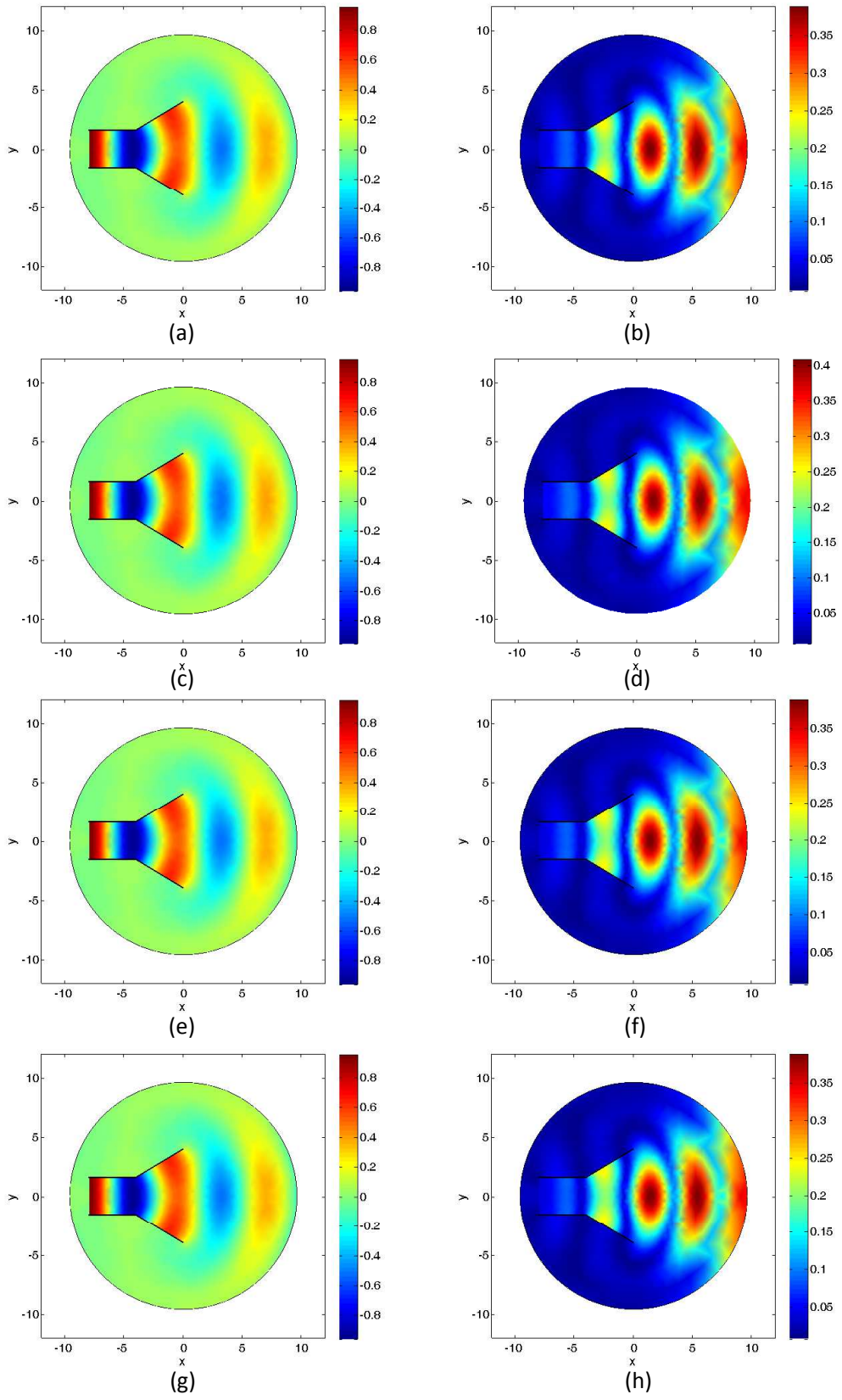

Figure 2.10: Horn (4 input dimensions): Comparison of the predictive variances using ALWPR with the MC estimates using 10^6 samples. The first row provides the MC mean (a) and the MC std (b). The next three rows are the predicted mean and predicted std with $\delta = 10^{-5}, 10^{-7}$ and 10^{-9}, respectively.

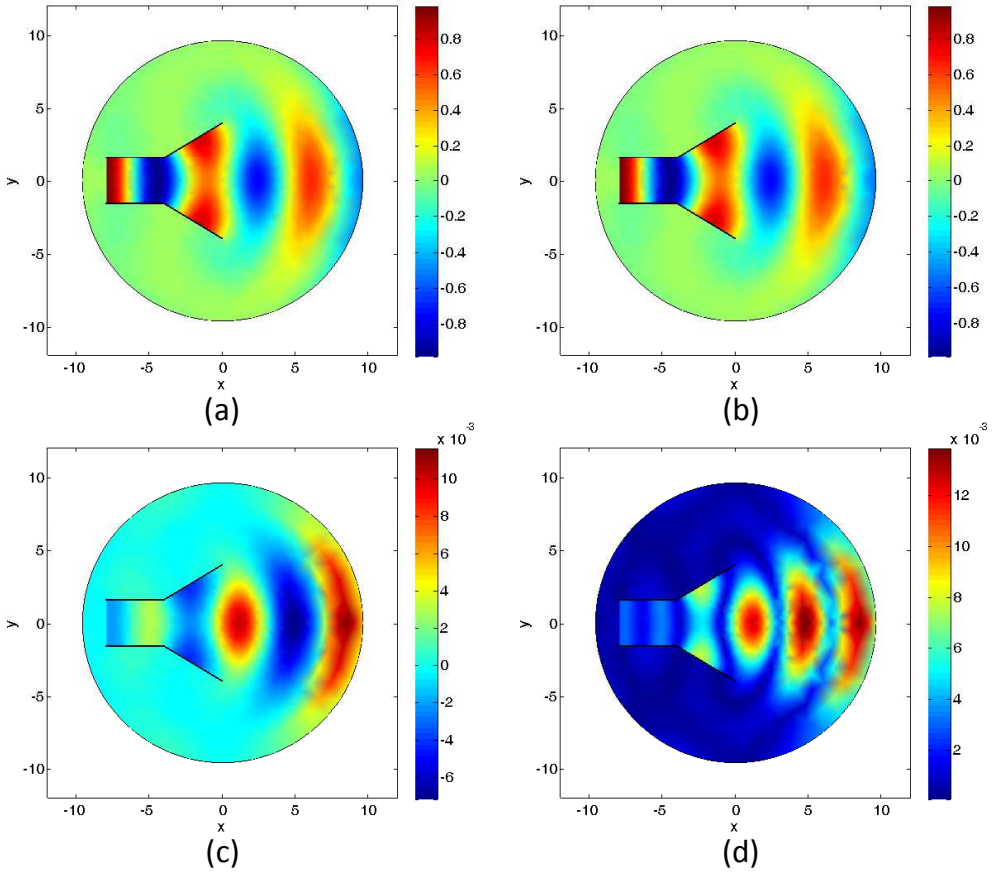

Figure 2.11: Horn (4 input dimensions): Comparison of the prediction at a random input point with the true response. (a) prediction given by ALWPR, (b) true response, (c) difference between the prediction and the true response, and (d) predictive variance given by ALWPR.

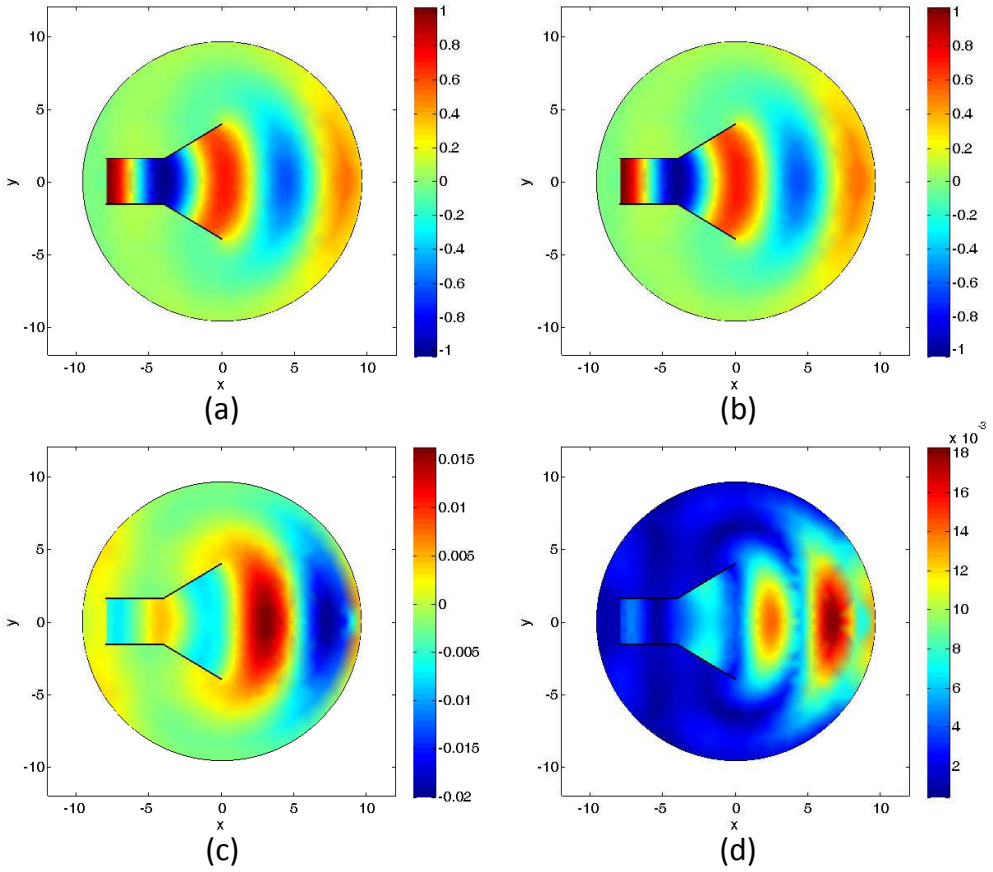

Figure 2.12: Horn (4 input dimensions): Comparison of the prediction at a random input point with the true response. (a) prediction given by ALWPR, (b) true response, (c) difference between the prediction and the true response, and (d) predictive variance given by ALWPR.

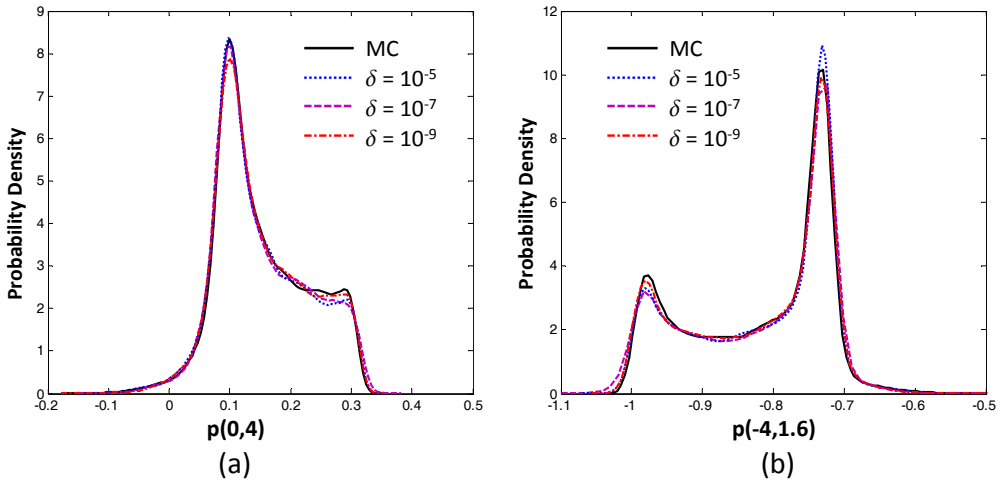

Figure 2.13: Horn (4 input dimensions): Comparison of the predictive PDF at two different spatial points using ALWPR with the corresponding MC predictions.

2.2.3 Elliptic Problem

In this section, we consider a benchmark stochastic elliptic problem:

$$
\begin{aligned}
-\nabla \cdot (a_K(\mathbf{x}, \cdot) \nabla u(\omega, \cdot)) &= f(\cdot), \text{ in } D, \\
u(\omega, \cdot) &= 0, \text{ on } \partial D,
\end{aligned}
\tag{2.28}
$$

where the physical domain is $D = [0, 1]^2$. In order to avoid confusion with the physical dimension $\mathbf{x} = (x, y)$, ω is used to denote the random variables instead of x. We choose a smooth deterministic load

$$
f(x, y) = 100 \cos(x) \sin(y), \tag{2.29}
$$

and work with homogeneous boundary conditions. The deterministic problem is solved with the finite element method using a 20×20 grid of bilinear quadrilateral elements. The random diffusion coefficient $a_K(\omega, \mathbf{x})$ is constructed as

$$
\log(a_K(\omega, x, y) - 0.5) = 1 + \omega_1 \left(\frac{\sqrt{\pi} L}{2} \right)^2 + \sum_{k=2}^{K} \xi_k \phi_k(x) \omega_k, \tag{2.30}
$$

where

$$
\xi_k := (\sqrt{\pi} L)^{1/2} \exp\left(-\frac{(\lfloor \frac{k}{2} \rfloor \pi L)^2}{8} \right), \text{ for k } \geq 2, \tag{2.31}
$$

and

$$
\phi_k(x) := \begin{cases} \sin\left(\frac{\lfloor \frac{k}{2} \rfloor \pi x}{L_p} \right), & \text{if } k \text{ is even,} \\ \cos\left(\frac{\lfloor \frac{k}{2} \rfloor \pi x}{L_p} \right), & \text{if } k \text{ is odd.} \end{cases} \tag{2.32}
$$

We choose $\omega_k,\ k = 1, \ldots, K$ to be independent identically distributed random variables

$$
\omega_k \sim \mathcal{B}eta([2, 5]). \tag{2.33}
$$

While this problem has been studied before with polynomial chaos and sparse grid approaches using uniform random variables, we select variables following

the $\mathcal{B}eta$ distribution in order to demonstrate the ability of the algorithm to bias the sample selection based on the input probability distribution. Hence, the stochastic input space is $\Omega = [0, 1]^K$. Finally, we set

$$L_p = \max\{1, 2L_c\} \text{ and } L = \frac{L_c}{L_p}, \tag{2.34}$$

where L_c is called the correlation length. The expansion Eq. (2.30) resembles the Karhunen-Loève expansion of a two-dimensional random field with stationary covariance

$$Cov[\log(a_K - 0.5)]((x_1, y_1), (x_2, y_2)) = \exp\left\{-\frac{(x_1 - x_2)^2}{L_c^2}\right\}. \tag{2.35}$$

In this example, we set the correlation length to $L_c = 0.6$ and study the problem with $K = 40$ input dimensions. The number of initial samples is chosen to be 1000. The convergence plots of the mean weighted predictive variance and L_2 norm of the error in variance are shown in Fig. 2.14. A comparison with the results obtained using the ASGC method and MC method is also shown. Fig 2.15 plots the predicted variance of the response for $K = 40$ against the MC estimate with 10^6 samples. Notice that as the threshold δ decreases, the predicted variance becomes indistinguishable from the MC estimates. Figs. 2.16 and 2.17 show the predictive capabilities of ALWPR for $K = 40$ at $\delta = 10^{-7}$ on two random input points. The predictions agree very well with the true responses. Also, by using 10^5 samples, we compare the predictive PDFs for two randomly selected outputs, $u(0.4, 0.15)$ and $u(0.5, 0.5)$, as shown in Fig. 2.18. It can be seen that the predicted PDFs are in good agreement with those obtained from MC. This example was run in parallel with 300 processors ($q = 200$ and $n = 100$).

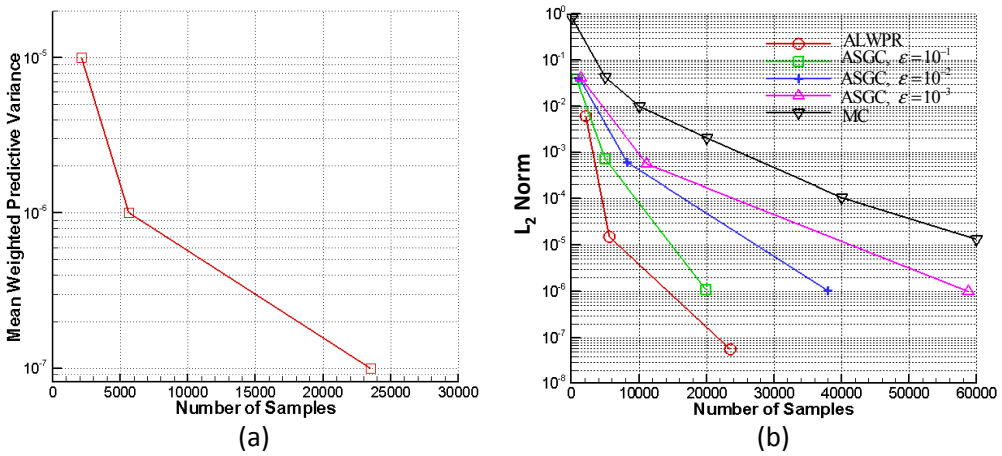

Figure 2.14: Elliptic example (40 input dimensions): (a) The mean weighted predictive variance as a function of the number of samples observed. (b) The L_2 norm of the error in variance as a function of the number of samples used by ALWPR, ASGC and MC.

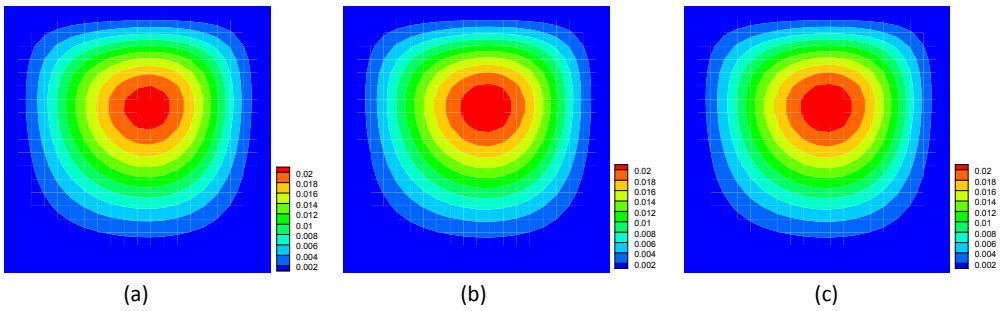

Figure 2.15: Elliptic example (40 input dimensions): Comparison of the predictive variances using ALWPR with (a) $\delta = 10^{-5}$, (b) $\delta = 10^{-7}$ and (c) a MC simulation using 10^6 samples.

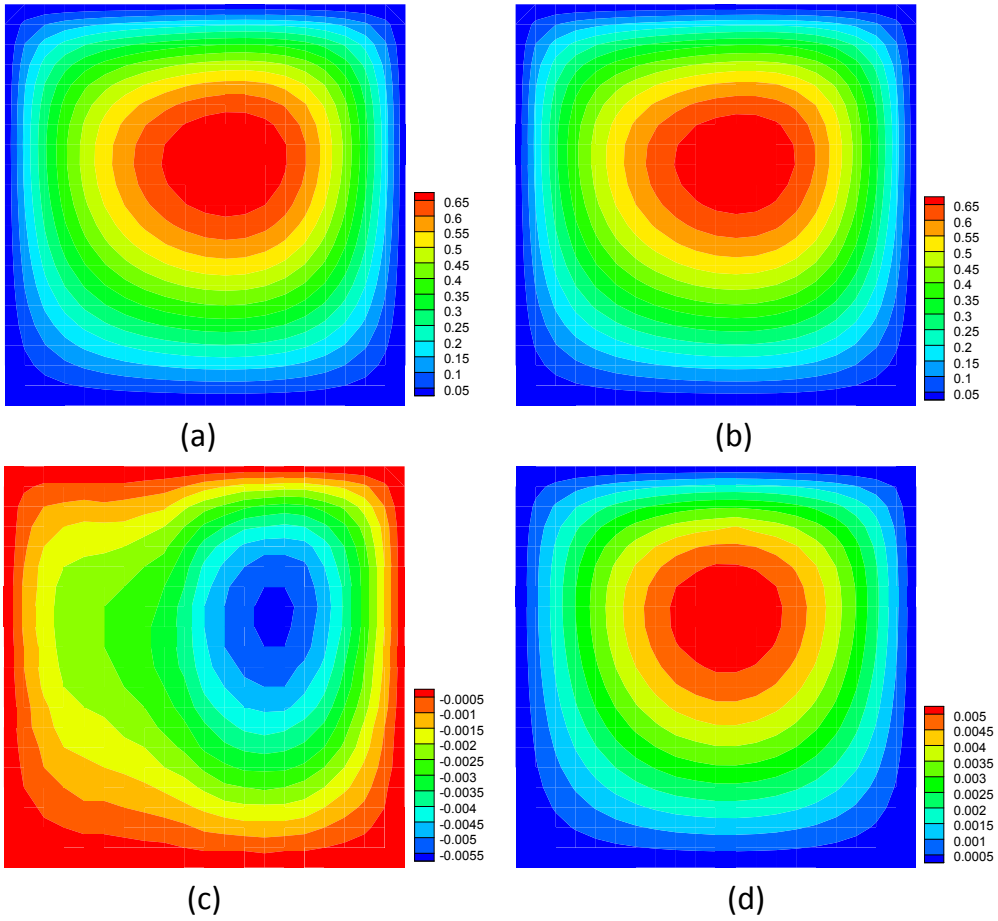

Figure 2.16: Elliptic example (40 input dimensions): Comparison of the prediction at a random input point with the true response. (a) Prediction given by the ALWPR, (b) True response, (c) Difference between the prediction and the true response and (d) Predictive variance given the ALWPR.

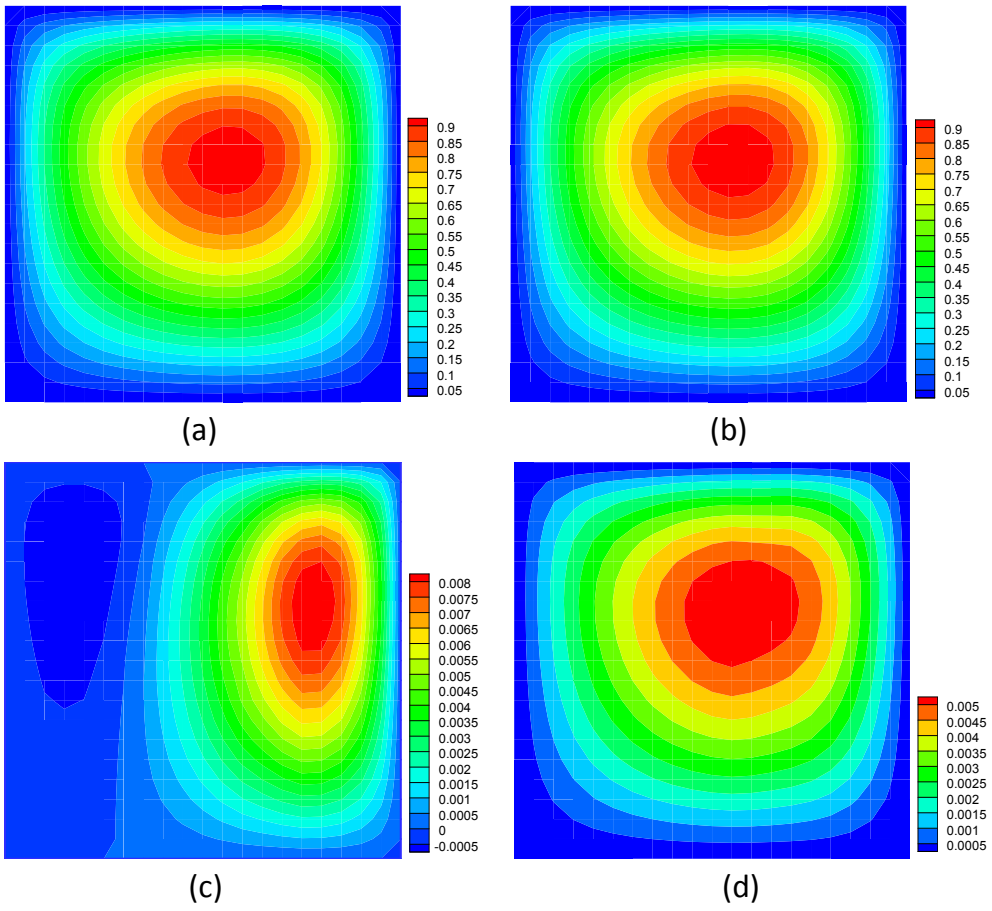

Figure 2.17: Elliptic example (40 input dimensions): Comparison of the prediction at a random input point with the true response. (a) Prediction given by the ALWPR, (b) True response, (c) Difference between the prediction and the true response and (d) Predictive variance given the ALWPR.

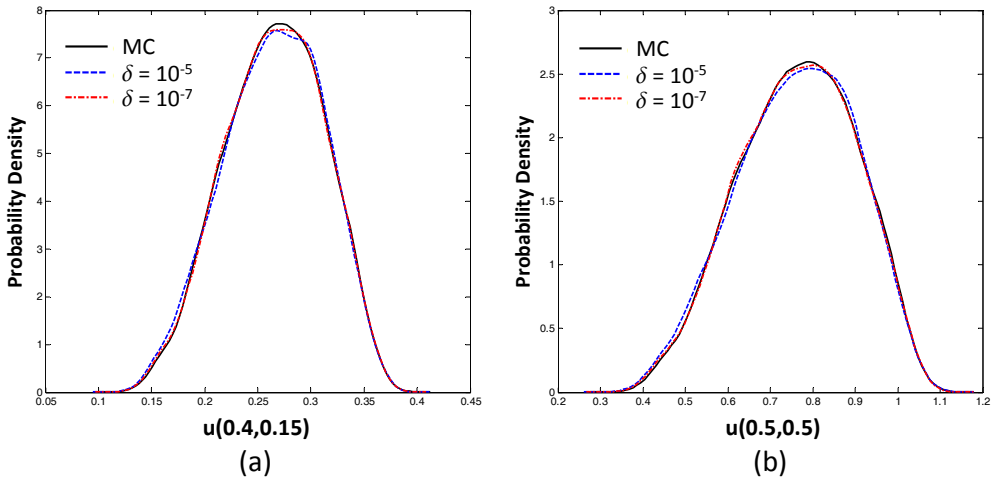

Figure 2.18: Elliptic example (40 input dimensions): Comparison of the predictive PDF at two different spatial points using ALWPR with the MC predictions.

2.3 Conclusions

An adaptive implementation of the locally weighted projection regression method was considered and applied to uncertainty quantification problems. The method works for any input distribution and provides predictions with error-bars at any query point. It can deal with multi-outputs and uses active learning in the selection of new sample input points. The selection of new input points is based on the predictive variance and an additional distance penalty term. Also, the method is capable of assigning a proper initial value of the distance metric for each local model depending on the local environment. Once the model is successfully built, it can provide rapid predictions at new query points thus making the ALWPR framework an inexpensive surrogate of the direct solver.

Various examples were considered to study the accuracy and efficiency of the developed ALWRP method. It was shown that the method is capable of predicting the correct statistics in the presence of discontinuities in the stochastic space. In the high-dimensional elliptic problem considered, the scheme captured well the first- and second-order statistics, and also provided reasonable predictions of the PDFs of the outputs. It is clear that at higher dimensions the performance of the method will be limited from issues related to the curse-of-dimensionality. The presented methodology treats multiple outputs in an independent fashion thus it cannot accurately predict correlations among them. This certainly can be a promising direction for expanding the framework. Finally, a complete Bayesian treatment of locally weighted progression regression if of current interest and work in this area will be reported in future works.

CHAPTER 3

A NONPARAMETRIC BELIEF PROPAGATION METHOD FOR UNCERTAINTY QUANTIFICATION WITH APPLICATIONS TO FLOW IN RANDOM POROUS MEDIA

This chapter is organized as follows. First, the problem definition is given in Section 3.1. Then the basic procedure of how to construct an appropriate graphical model and all the associated algorithms are discussed in Sections 3.2, 3.3 and 3.4. In Section 3.5, we introduce the porous media flow problem and provide various examples demonstrating the efficiency and accuracy of the graphical model approach. Brief discussion and conclusions are finally provided in Section 3.6.

3.1 Problem definition

Consider a random field $\{A_{\mathbf{x}}\}_{\mathbf{x}\in D}$, where $D \subset \mathbb{R}^d, d = 1, 2, 3$ is the spatial domain of interest (physical space), $\mathbf{x} = (x_1, \ldots, x_d) \in D$ is a spatial point. We think of a realization of this random field as being the input to a deterministic solver that models a physical problem of interest. In this way, we may define the response random field $\{Y_{\mathbf{x}}\}_{\mathbf{x}\in D}$. In particular, we investigating the problem of flow through random porous media. In this case, the input random field is the rock permeability field.

In practice, the physical domain is decomposed into fine-elements on which the permeability is defined. In particular, consider a partition, $\mathcal{T}_f$, of the domain D in N_f non-overlapping elements e_i, i.e., $\mathcal{T}_f = \bigcup_{i=1}^{N_f} e_i$. The random input field is approximated in a piece-wise linear fashion over the fine grid. We denote the

resulting random vector by $\mathbf{A}$, where:

$$\mathbf{A} = (A_1, A_2, \ldots, A_{N_f}). \tag{3.1}$$

Usually, we are only interested in the responses on a coarser grid than $\mathcal{T}_f$. Therefore, let us define a coarser partition of the same domain D. Denote this partition as $\mathcal{T}_c = \bigcup_{i=1}^{N_c} E_i$, where N_c is the number of coarse-elements. Fig. 3.1 shows a fine-grid (finer lines) and a corresponding coarse-grid (heavier lines). Let N_G denote the number of nodes on the coarse-grid. The response field is approximated by the random vector of responses on the coarse nodes:

$$\mathbf{Y} = (Y_{\mathbf{x}_1}, Y_{\mathbf{x}_1}, \ldots, Y_{\mathbf{x}_{N_G}}). \tag{3.2}$$

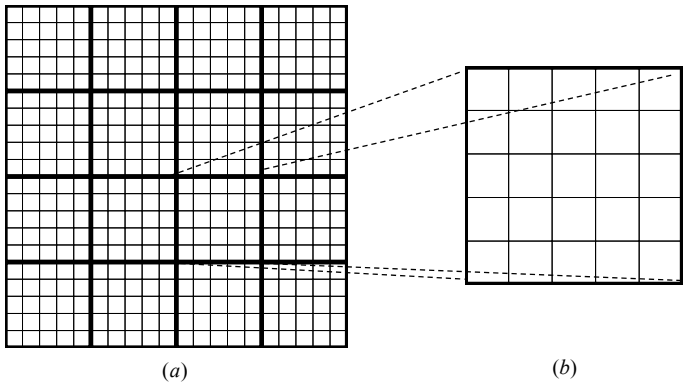

Figure 3.1: Schematic of the domain partition: (a) fine- and coarse-scale grids and (b) fine-scale local region in one coarse-element.

Consider the multi-output nonlinear function $\mathbf{f} : \mathbb{R}^{N_f} \rightarrow \mathbb{R}^{N_G}$ modeling the physical problem of interest (deterministic solver), i.e., $\mathbf{Y} = \mathbf{f}(\mathbf{A})$. In uncertainty quantification tasks, one specifies a probability density on the input $\mathbf{A}$, $p(\mathbf{A})$, and is interested in quantifying the probability measure induced by it on the response. Formally, the marginal distribution of the responses $\mathbf{Y}$ given the deterministic solver $\mathbf{f}(\cdot)$ can be obtained by:

$$p(\mathbf{Y}|\mathbf{f}(\cdot)) \quad = \quad \int \delta(\mathbf{Y} - \mathbf{f}(\mathbf{A}))p(\mathbf{A})d\mathbf{A}$$

$$= \int p(\mathbf{Y}|\mathbf{A}, \mathbf{f}(\cdot))p(\mathbf{A})d\mathbf{A}. \tag{3.3}$$

The above equation is based on the knowledge of the stochastic input model and the dependence between output and input. In theory, the mapping from $\mathbf{A}$ to $\mathbf{Y}$ is deterministic given the deterministic model $\mathbf{f}(\cdot)$. In this work, we learn this relationship between input and output completely from training data set $\mathcal{D} = \{\mathbf{A}^{(n)}, \mathbf{Y}^{(n)}\}$. Our state of knowledge after observing the simulations $\mathcal{D}$, is neatly captured in a Bayesian way by $p(\mathbf{Y}|\mathbf{A}, \mathcal{D})$, i.e. conditioning on the data instead of the solver. Eq. (3.3) can now be replaced by

$$p(\mathbf{Y}|\mathcal{D}) = \int p(\mathbf{Y}|\mathbf{A}, \mathcal{D})p(\mathbf{A})d\mathbf{A}. \tag{3.4}$$

The challenging part of representing $p(\mathbf{Y}|\mathbf{A}, \mathcal{D})$ is due to the high-dimensionality of $\mathbf{A}$. We deal with this issue by: 1) Reducing the dimensionality of $\mathbf{A}$; 2) Localizing the connections of $\mathbf{A}$ and $\mathbf{Y}$. These developments are is discussed in the next section.

3.2 Model reduction

Let us assume that a set of realizations of the random input vector $\mathbf{A}$ are given. Using this data in conjunction with the Empirical Karhunen-Loève expansion [39], we construct a reduction map $\mathcal{R}_g$:

$$\boldsymbol{\xi} = \mathcal{R}_g(\mathbf{A}), \tag{3.5}$$

where ξ is a reduced set of variables. In [64, 9], the authors constructed different models to build a map from ξ to $\mathbf{Y}$. The uncertainty propagation problem in Eq. (3.4) can now be re-formulated as:

$$p(\mathbf{Y}|\mathcal{D}) = \int p(\mathbf{Y}|\boldsymbol{\xi}, \mathcal{D})p(\boldsymbol{\xi})d\boldsymbol{\xi}. \tag{3.6}$$

where $p(\xi) = \int \delta(\xi - \mathcal{R}_g(\mathbf{A}))d(\mathbf{A})\mathbf{A}$. However, even ξ has less dimension than the real input $\mathbf{A}$, it is still difficult to capture $p(\mathbf{Y}|\xi, \mathcal{D})$. Therefore, we propose to a way to localize the problem. The underlying assumption we made is a physically reasonable assumption for many problems that the response at one coarse-grid node correlates strongly on the input permeability in the underlying nearest coarse-elements, and that the influence by the permeability at all other coarse-elements can be ignored, as shown in Fig. 3.2.

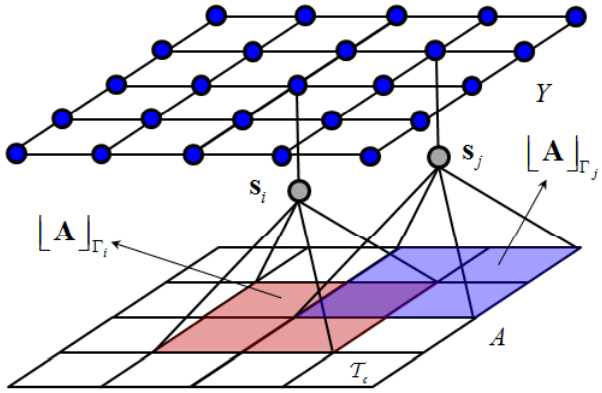

Figure 3.2: An illustration of the model reduction framework considered in this paper. The response at each coarse-node depends on the permeability field at the neighboring coarse-elements.

Let $\Gamma_i \subset \{1, \ldots, N_c\}$ be the set of neighboring coarse-elements corresponding to the coarse-node i. Let $\lfloor \mathbf{A} \rfloor_{\Gamma_i}$ denote the input vector over the neighboring coarse-elements close to coarse-node i (see Fig. 3.2), where $\lfloor \cdot \rfloor$ is the restriction operator. Based on the given realizations of the input vector $\lfloor \mathbf{A} \rfloor_{\Gamma_i}$, one can perform a Karhunen-Loève expansion and obtain the reduced representation $\mathbf{s}_i$ of the permeability field over Γ_i that encodes most of the information relevant to the response at coarse-node i as:

$$\mathbf{s}_i = \mathcal{R}_{\Gamma_i}(\lfloor \mathbf{A} \rfloor_{\Gamma_i}), \tag{3.7}$$

where $\mathcal{R}_{\Gamma_i}$ is the reduction map for the input in the elements Γ_i. $\mathbf{s}_i$ should be

correlated to its neighboring reduced representation $\mathbf{s}_j$ due to the overlapped inputs considered, as shown in Fig. 3.2.

The connection between this local model reduction and the global model reduction in Eq. (3.5) is given as follows: Let C_g be the global reconstruction map such that:

$$\mathcal{R}_g(C_g(\boldsymbol{\xi})) = \boldsymbol{\xi}, \tag{3.8}$$

and let C_{Γ_i} be the local reconstruction map corresponding to the model reduction for the i^{th} coarse-grid node defined as:

$$\lfloor \widetilde{\mathbf{A}} \rfloor_{\Gamma_i} = C_{\Gamma_i}(\mathbf{s}_i), \tag{3.9}$$

where $\lfloor \widetilde{\mathbf{A}} \rfloor_{\Gamma_i}$ is the reconstructed local random field on the coarse-elements Γ_i.

We can write the following:

$$\mathbf{s}_i \approx \mathcal{R}_{\Gamma_i}([C_g(\boldsymbol{\xi})]_{\Gamma_i}), \tag{3.10}$$

where $[\cdot]_{\Gamma_i}$ is the restriction of $\widetilde{\mathbf{a}} = C_g(\xi)$ over Γ_i. Similarly, we can write,

$$\boldsymbol{\xi} \approx \mathcal{R}_g\{\mathcal{H}[C_{\Gamma_1}(\mathbf{s}_1), \ldots, C_{\Gamma_{N_G}}(\mathbf{s}_{N_G})]\}, \tag{3.11}$$

where $\mathcal{H}$ is a function that approximates the global reconstruction function C_g using all input realizations obtained from local reduction models. Note that the local reconstructions have overlaps, an example of overlapped region of input is shown in Fig. 3.2, as the overlapped region of the red square and blue square. The above equation defines how the local reduced input $\mathbf{s}_i$ is correlated to ξ through the reduction/reconstruction maps. The better the choice of $\mathcal{H}$ is, the closer the local/global approximations we obtain. In this work, we simply take the average of the all the reconstructions over the overlapped regions, so the function $\mathcal{H}$ is given as:

$$\lfloor H(\cdot)\rfloor_{E_i} = \frac{1}{N_{\text{overlap}}} \sum_{j=1}^{N_{\text{overlap}}} \lfloor C_{\Gamma_j}(\mathbf{s}_j)\rfloor_{E_i}, \tag{3.12}$$

where N_{overlap} is the number of reconstructions over the overlapped regions by different local reduction models over coarse-element E_i, and $\lfloor\cdot\rfloor_{E_i}$ is the restriction of $\widetilde{\mathbf{a}}_{\Gamma_j} = C_{\Gamma_j}(\mathbf{s}_j)$ over the E_i element.

Remark 1**:** The reduced variables $\mathbf{s}_i$ affiliated with different locations i are different random variables. For a stationary permeability random field, local features have the same distribution on coarse-elements, therefore, the localized reduced random variables $\mathbf{s}_i$ are going to follow the same distribution for all i (not considering boundary effects). However, notice that one can not say these $\mathbf{s}_i$ are the same random variables even if they follow the same distribution. Given a realization of stationary stochastic input $\mathbf{a}^{(n)}$, the local features $\mathbf{a}_k^{(n)}$ and $\mathbf{a}_l^{(n)}$ on coarse-elements E_k and E_l are in general different. For nonstationary random permeability field, the $\mathbf{s}_i$ variables at different locations i follow different marginal distributions.

The localization of the input model reduction outlined above can be implemented with literally any model reduction technique including linear and nonlinear dimension reduction algorithms. For linear dimension reduction, the most famous and the most widely used method is the Principal Component Analysis (PCA) [115] method. The first version of PCA method appeared half a century ago and it has been shown since then to be a reliable reduction method forming the basis of many other more advanced mathematical reduction methodologies. In the last decade, a large number of nonlinear dimensional reduction techniques have been proposed (e.g., [20, 89, 111]). Most of the nonlinear techniques are not as well studied and have been shown often

to provide better performance than PCA for artificial than physical nonlinear datasets [103]. These methods perform not better (sometimes much poorer) than PCA for natural datasets [103]. Therefore, here for simplicity of the presentation, we choose PCA as the dimension reduction technique. This will allow us to emphasize the graph theoretic approach for solving the underlying stochastic flow problem of interest.

Up to now, we have completely defined the relationship between ξ and $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_{N_G}\}$, which means we don't need the complete input information anymore. Given the distribution of ξ, it is straightforward to use Monte Carlo method to find the distribution of $\mathbf{S}$, $p(\mathbf{S})$. The map from $\mathbf{A}$ to $\mathbf{Y}$ can be now reexpressed by the map from $\mathbf{S}$ to $\mathbf{Y}$. Then computing $p(\mathbf{Y}|\mathcal{D})$ requires to compute the conditional $p(\mathbf{Y}|\mathbf{S}, \mathcal{D})$, instead of $p(\mathbf{Y}|\mathbf{A}, \mathcal{D})$. Indeed, we can write the following:

$$\begin{aligned} p(\mathbf{Y}|\mathcal{D}) &= \int p(\mathbf{Y}|\mathbf{A}, \mathcal{D}) p(\mathbf{A}) d\mathbf{A} \\ &\approx \int p(\mathbf{Y}|\mathbf{S}, \mathcal{D}) p(\mathbf{S}|\xi, \mathcal{D}) p(\xi) d\mathbf{S} d\xi. \end{aligned} \tag{3.13}$$

As discussed in Chapter 1, probabilistic graphical models [58] can be used to systematically explain the probabilistic relationship between $\mathbf{S}$ and the responses $\mathbf{Y}$. Their joint distribution can be partitioned in a way the accounts for the local nature of the dependence/correlation of the response to the input variables. The details of such approach are introduced in Section 3.3.

3.3 Probabilistic graphical model

We are given a number of realizations of the global reduced input random variables ξ, localized reduced input random variables $\mathbf{S}$ and corresponding responses $\mathbf{Y}$, and also the input distribution of $p(\xi)$. The relationship between ξ and $\mathbf{S}$ is deterministic, therefore, our main objective is building a probabilistic graphical model between $\mathbf{S}$ and $\mathbf{Y}$, based on the given set of realizations. We next plan to use an inference algorithm on the probabilistic graph to address uncertainty quantification problems.

3.3.1 Brief introduction to probabilistic graphical models

A graphical model aims to represent the joint probability distribution of many random variables efficiently by exploiting factorization [58]. The two most common forms of graphical models are directed graphical models and undirected graphical models, based on directed graphs and undirected graphs, respectively. The dependence relationship is visible directly from the graph for the directed graph model, while the dependence relationship is hidden in the undirected graph. In this work, the dependence relationships between the response random variables are not clear, so we focus on the undirected graph, which is also called pairwise Markov Random Field (MRF) [56].

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ are the nodes (random variables) and $\mathcal{E}$ are the edges of the graph (correlations). Let $\{X_{\mathcal{V}} : x_i \in \mathcal{V}\}$ be a collection of random variables indexed by the nodes of the graph and let C denote a collection of cliques of the graph (i.e., fully connected subsects of nodes). Asso-

ciated with each clique $c \in C$, let $\phi_c(X_c)$ denote a nonnegative potential function, which implicitly encodes the dependence information among the nodes within the clique. The joint probability $p(X_\mathcal{V})$ is defined by taking the product over these potential functions and normalizing,

$$p(X_\mathcal{V}) = \frac{1}{Z} \prod_{c \in C} \phi_c(X_c), \tag{3.14}$$

where Z is a normalization factor.

The graphical model representation makes the inference problem easier. The general algorithm of probabilistic inference is that of computing the marginal probability $p(\mathbf{X}_\mathcal{H})$ or conditional probability $p(\mathbf{X}_\mathcal{H}|\mathbf{X}_\mathcal{O})$, where $\mathcal{V} = \mathcal{O} \cup \mathcal{H}$ for given subsets $\mathcal{O}$ and $\mathcal{H}$. The belief propagation algorithm (inference) is then applied to find the marginal or conditional probabilities of interest. Notice if an event $\{X_\mathcal{O} = \mathbf{x}_\mathcal{O}\}$ is observed, the original clique potentials need to be modified, that is, for $\{X_i, i \in \mathcal{O}\}$, we multiply the potential $\phi_c(X_c)$ by the Kronecker Delta function $\delta_{X_i}(\mathbf{x}_i)$ for any clique $c \in C$ such that $\{i \in c \cap \mathcal{O}\}$, where $\mathbf{x}_i$ is the observation of node i. The detailed inference algorithm will be discussed in Section 3.4.

3.3.2 The structure of the graph

To find an efficient structure of the graph, let us start from the joint distribution of $(\mathbf{Y}, \mathbf{S}, \xi|\mathcal{D})$,

$$\boldsymbol{p(\mathbf{Y}, \mathbf{S}, \xi|\mathcal{D}) = p(\mathbf{Y}|\mathbf{S}, \mathcal{D})p(\mathbf{S}|\xi, \mathcal{D})p(\xi).} \tag{3.15}$$

The above decomposition is based on the assumption that $\mathbf{S}$ contains all information ξ contains for the calculation of $\mathbf{Y}$. The probabilistic relationship between $\mathbf{S}$ and ξ is discussed in Section 3.2. Each variable $\mathbf{s}_i \in \mathbf{S}$ has a deterministic

relationship with ξ, therefore, all the $\mathbf{s}_i$'s should be directly linked with ξ. The correlation among the $\mathbf{s}_i$ variables is then reflected via their connections with ξ. The corresponding structure between $\mathbf{S}$ and ξ is given in Fig. 3.3.

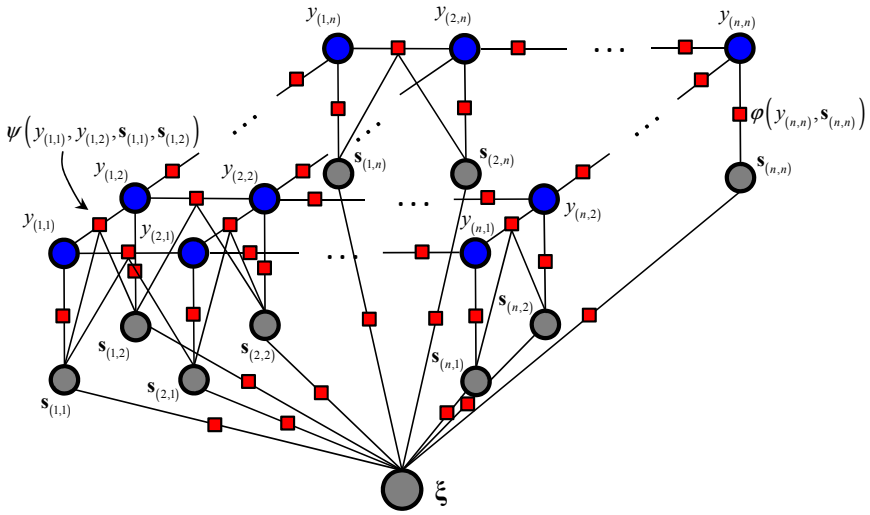

Figure 3.3: The general graph structure for the problem of interest. The y variables represent the response of the system (velocities and/or pressure on a coarse-grid), ξ represents the reduced set of random variables defining the random permeability over the whole domain D and $\mathbf{s}_{(i,j)}$ is the reduced set of random variables defining the random permeability on the patch of coarse-elements that share the coarse-node (i, j). Note here in this two-dimensional framework, we identify our nodes with two indices (i, j) rather than the single indices $1, 2, \ldots, N_G$ used before. The red squares are the factor nodes that represent the potentials.

To find the structure between $\mathbf{Y}$ and $\mathbf{S}$, we need to find an approximate decomposition of $p(\mathbf{Y}|\mathbf{S}, \mathcal{D})$ in Eq. (3.13),

$$p(\mathbf{Y}|\mathbf{S}, \mathcal{D}) = p(y_1, y_2, \ldots, y_{N_G}|\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{N_G}, \mathcal{D}). \tag{3.16}$$

In this work, we consider only the pairwise correlations between the response random variables y, among which, only correlations between neighboring response variables are considered. So the conditional distribution $p(\mathbf{Y}|\mathbf{S}, \mathcal{D})$

can be decomposed as

$$p(\mathbf{Y}|\mathbf{S},\mathcal{D}) \approx \prod_{i=1}^{N_G} p(y_i|\mathbf{S},\mathcal{D}) \prod_{j\in\Gamma(i)} p(y_i,y_j|\mathbf{S},\mathcal{D}), \tag{3.17}$$

where $\Gamma(i)$ denotes the set of neighboring nodes of node i.

Remark 2**:** This decomposition is inspired by the general treatment to the conditional random field representation of a Gibbs distribution, where in principle the explicit expansion of the conditional distribution involves one-body term and two-body interaction terms to n-body interaction term. In practice, we ignore higher-order interactions and only keep the first two terms [106, 81]. In [108], the authors also apply a similar idea in factorizing the complex conditional probability distribution.

To further simplify the dependencies in the factorization of $p(\mathbf{Y}|\mathbf{S},\mathcal{D})$, we further assume that the response at one coarse-grid node is strongly dependent on the inputs in its underlying nearest coarse-elements, thus the influence by all other inputs is ignored. In other words, we are assuming that y_i only depends on its underlying localized reduced input $\mathbf{s}_i$. This is in analogy to various multiscale methods (e.g. the MsFEM method [30]) where in the calculation of the local multiscale basis functions only the local permeability is considered. In addition, we assume that the correlations between the physical responses are only dependent on the affiliated local features, which is different from the work in [108], where the authors assumes the correlation terms depend on the whole input field. Hence, the above equation can be further decomposed as

$$p(\mathbf{Y}|\mathbf{S},\mathcal{D}) \approx \prod_{i=1}^{N_G} p(y_i|\mathbf{s}_i,\mathcal{D}) \prod_{j\in\Gamma(i)} p(y_i,y_j|\mathbf{s}_i,\mathbf{s}_j,\mathcal{D}). \tag{3.18}$$

The constructed structure of the undirected graph is shown in Fig. 3.3. If we

write Eq. (3.18) as a product of potential functions in the graphical model, we can obtain

$$p(\mathbf{Y}|\mathbf{S},\mathcal{D}) \propto \prod_{k\in\mathcal{V}^{(\mathbf{y})}} \varphi_k(y_k,\mathbf{s}_k) \prod_{(i,j)\in\mathcal{E}^{(y)}} \psi_{i,j}(y_i,y_j,\mathbf{s}_i,\mathbf{s}_j). \tag{3.19}$$

There are two kinds of potential functions in Eq. (3.19), one is cross potential functions, $\varphi(*)$, which interprets the relationship between the reduced input variables $\mathbf{s}_i$ and response variables y_i; the other one is correlation potential functions, $\psi(*)$, that model the correlation between neighboring response variables. In the potential functions, the unknown parameters will be learned by using the training data $\mathcal{D}$, the learning process is discussed in Section 3.3.3. In Eq. (3.19), we only put the affiliated random variables in the bracket, and we omit $\mathcal{D}$ for math convenience. In the following, we denote with $\mathcal{V}^{(\mathbf{s})}$ the set of the localized reduced input nodes (coarse-grid nodes), and with $\mathcal{E}^{(y)}$ the set of the edges between the response variables (edges of the coarse-elements).

In this work, the potential functions between $\mathbf{s}_i$ and ξ are difficult to model due to their high dimensionality nature. However, $\mathbf{S}$ and ξ are explicitly known to us, so is the relationship between them. Therefore, we do not have to learn these potential functions explicitly. In Section 3.4, we will discuss how we perform the inference problem without using the potential functions between $\mathbf{s}_i$ and ξ.

3.3.3 Learning the graphical model

Since we are considering a nonparametric graphical model, the potential functions should have Gaussian mixture forms. As discussed above, there are two types of potential functions, the cross potential function for response variables

and localized reduced input variables, $\varphi(*)$, and the correlation potential function for the response variables, $\psi(*)$, as given in Eq. (3.19). In this work, both potential functions are designed to have the following form,

$$\psi_{i,j}(z_i, z_j) = \sum_{m=1}^{M} \omega^{(m)} \mathcal{N}\left((z_i, z_j); \mu^{(m)}, \Sigma^{(m)}\right), \tag{3.20}$$

where z_i and z_j denote the random variables on node i and node j and $\mathcal{N}(\cdot)$ is the Gaussian distribution. The unknown parameters in the above potential functions are $\{\omega^{(m)}, \mu^{(m)}, \Sigma^{(m)}, m = 1, \ldots, M\}$, where $\omega^{(m)}$ is the weight (scalar) for component m, $\mu^{(m)}$ is the mean for component m (the size of the mean vector is equal to the sum of dimensions of z_i and z_j), and $\Sigma^{(m)}$ is the covariance matrix.

In this work, the unknown parameters in the potential functions are learned by maximizing the log-likelihood. Denote $\boldsymbol{\Theta} = \left\{\boldsymbol{\theta}_{i,j} : (i,j) \in \mathcal{E}^{(y)} \text{ and } \boldsymbol{\theta}_i : i \in \mathcal{V}^{(y)}\right\}$ as the set of all the unknown parameters, where $\theta_{i,j} = \left\{\omega_{i,j}^{(m)}, \mu_{i,j}^{(m)}, \Sigma_{i,j}^{(m)}; m = 1, \ldots, M\right\}$ and $\theta_i = \left\{\omega_i^{(m)}, \mu_i^{(m)}, \Sigma_i^{(m)}; m = 1, \ldots, M\right\}$. These parameters can be calculated locally in the coarse-grid. For specific i, j such that $i \in \mathcal{V}^{(y)}$ and $(i,j) \in \mathcal{E}^{(y)}$, let us consider given N observations of $\{(\mathbf{s}_i^{(n)}, \mathbf{s}_j^{(n)}), (y_i^{(n)}, y_j^{(n)})\}$ for $n = 1, \ldots, N$. The log-likelihood can then be calculated as,

$$\mathcal{L}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i,j} | \mathcal{D}) = \sum_{n=1}^{N} \left[\log p\left(\mathbf{y}_i^{(n)}, \mathbf{s}_i^{(n)} | \boldsymbol{\theta}_i\right) + \log p\left(\mathbf{s}_i^{(n)}, \mathbf{s}_j^{(n)}, \mathbf{y}_i^{(n)}, \mathbf{y}_j^{(n)} | \boldsymbol{\theta}_{i,j}\right)\right]. \tag{3.21}$$

By maximizing the log-likelihood, we obtain

$$(\widehat{\boldsymbol{\theta}_i}, \widehat{\boldsymbol{\theta}_{i,j}}) = \arg\max_{\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i,j}} \mathcal{L}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i,j} | \mathcal{D}). \tag{3.22}$$

Notice that maximizing the log-likelihood is equivalent to maximizing each component of Eq. (3.21) separately, therefore, the graph learning problem can be divided into a number of local learning problems. For example, to learn θ_i,

we only need to maximize $\sum_{n=1}^{N} \log p(y_i^{(n)}, \mathbf{s}_i^{(n)}|\theta_i)$ using the local training data set $\{y_i^{(n)}, \mathbf{s}_i^{(n)}; i = 1, \ldots, N\}$.

Remark 3**:** The parameters $\theta_{i,j}$ define the correlation between the response variables, whereas θ_i interpret the dependence relation between a response variable and its underlying localized reduced random input. Both of these parameters are computed locally using the training data. Thus the computational cost affiliated with the estimation of the parameters that define the probabilistic dependencies in the graph is minimal. Note that the approach used here is different from that in [108] where local estimation problems are only posed to compute the dependencies on the input permeability permeability of the local potentials. In this work, the effect of the input permeability is introduced via the local random variables $\mathbf{s}_i$ and the potentials considered are Gaussian mixtures with unknown parameters. The potentials in [108] are simple Gaussians. In general case, all the $\theta_{i,j}$ and θ_i are different across the graph (i.e. for different i and j).

Remark 4**:** For a stationary permeability case, the variables $\mathbf{s}_i$ follow the same distribution but this does not imply that θ_i are the same parameters. Taking simultaneous realizations of $\mathbf{s}_i$ and $\mathbf{s}_j$ leads to different local permeability realizations and thus different response fields $y_i^{(n)}$ and $y_j^{(n)}$, $n = 1, \ldots, N$. In the calculation of θ_i, the training data set $\{y_i^{(n)}, \mathbf{s}_i^{(n)}, i = 1, \ldots, N\}$ that we use vary with the location i and therefore θ_i differs with location. Similar argument can be made about the location dependence of the parameters $\theta_{i,j}$.

The Expectation Maximization (EM) algorithm is chosen to maximize the local log-likelihood. Note that in the EM algorithm employed, the number of mixture components M is predefined. A discussion of how to choose M is pro-

vided in Section 3.5.1.

3.4 Inference problem

The general inference problem in a graphical model is to find the marginal or conditional probabilities of interest in the graph. This task is usually performed by using the belief propagation (BP) algorithm [58]. In this work, after all the unknown parameters in the graph are successfully learned, the inference problem can be performed. In the following, we first provide a brief introduction to the general belief propagation algorithm, and then we discuss in detail how to apply the belief propagation algorithm into our framework.

3.4.1 General belief propagation

Belief propagation (BP) is a general inference algorithm for graphical models [56]. In the BP algorithm, each node iteratively solves the global inference problem by integrating information from the local environment, and then transmits a summary message to all its neighbors along the edges. The information flow during this process is called the message, or belief, which is a function containing sufficient information of the "influence" that one variable exerts on another.

Consider a general factor graph in Fig. 3.4. Let $\Gamma(y_q)$ denote all the factor nodes directly linked to variable y_q. At iteration t of the BP algorithm, the message from y_q to factor node f is a function of y_q, as shown in Fig. 3.4(a), the

update rule [56, 58] is

$$m^{(t)}_{y_q \to f}(y_q) \leftarrow \prod_{f_{pq} \in \Gamma(y_q) \setminus f} m^{(t-1)}_{f_{pq} \to y_q}(y_q). \tag{3.23}$$

Let us use $\mathcal{Y}_f$ to denote neighboring variables directly linked to factor node f, the message from a factor node f to variable y_q is also a function of y_q which is recursively updated by

$$m^{(t)}_{f \to y_q}(y_q) \leftarrow \int_{\mathcal{Y}_f \setminus y_q} f(\mathcal{Y}_f) \prod_{y_r \in \mathcal{Y}_f \setminus y_q} m^{(t)}_{y_r \to f}(y_r) d\mathcal{Y}_f \setminus y_q. \tag{3.24}$$

When the factor graph contains loops, the messages must be updated iteratively until convergence is achieved. Although until now there is no strict mathematical justification that the loopy belief propagation converges to the true marginals, in many applications, the resulting LBP algorithm exhibits excellent performance [94, 35, 114, 71]. Recently, several theoretical studies have provided insights into the approximations made by LBP, establishing connections to other variational inference algorithms and partially justifying its application to graphs with cycles [107, 112]. An estimate of the posterior marginal distribution of y_q at each iteration is obtained by gathering all the messages coming from neighboring factor nodes [56, 58]:

$$p^{(t)}(y_q) \propto \prod_{f \in \Gamma(y_q)} m^{(t)}_{f \to y_q}(y_q). \tag{3.25}$$

3.4.2 Inference approach for the problem of interest

In this section, suppose $p(\xi)$ is known. The objective is then to find the posterior marginal distribution of $p(\mathbf{Y}|\mathcal{D})$ via the belief propagation algorithm. In the

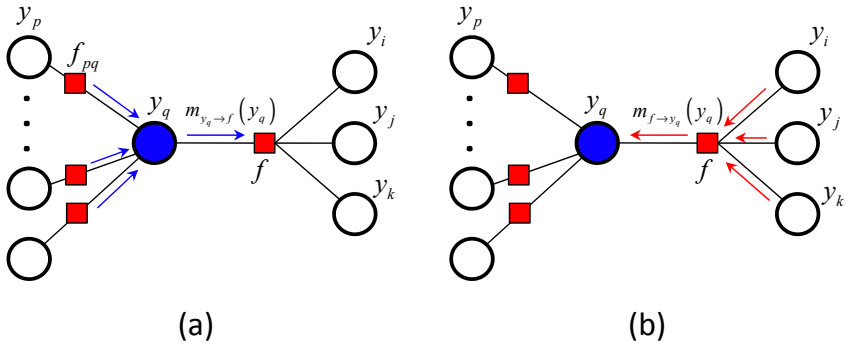

Figure 3.4: Message-passing recursions in a factor graph: (a) message passing from a variable node to a factor node, (b) message passing from a factor node to a variable node.

particular problem of interest, there are two main challenges with regards to the inference algorithm: (1) how to represent and calculate the message from ξ to $\mathbf{S}$ (discussed in Section 3.4.2), and (2) how to calculate the message update in the form of a Gaussian mixture (discussed in Section 3.4.2).

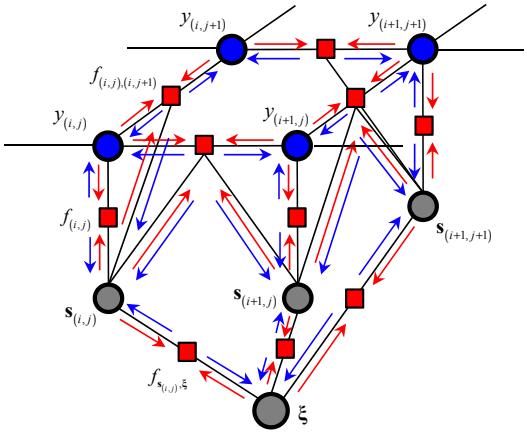

Figure 3.5: Message flow in the present graphical model framework. We assume a two-dimensional response with response variables (velocity components/pressure indicated by the blue nodes).

Detailed inference algorithm

The illustration of the message flow in the current graphical model framework is given in Fig. 3.5. Note that in this two-dimensional framework, we identify our nodes with two indices (i, j) rather than the single indices $1, 2, \ldots, N_G$ used in our earlier analysis. From the graph, we can see that there are three kinds of variable nodes in the graph, response nodes $y_{(i,j)}$, localized reduced input random variable nodes $\mathbf{s}_{(i,j)}$ and the global input random variable node ξ. Furthermore, there are three kinds of factor nodes in our framework, correlation factor nodes $f_{(i,j),(k,l)} = \psi(y_{(i,j)}, y_{(k,l)}, \mathbf{s}_{(i,j)}, \mathbf{s}_{(k,l)})$, cross factor nodes $f_{(i,j)} = \varphi(y_{(i,j)}, \mathbf{s}_{(i,j)})$, and the factor nodes that link $\mathbf{s}_{(i,j)}$ and ξ, $f_{\mathbf{s}_{(i,j)},\xi}$, which in practice don't have to be known explicitly. As discussed in the previous section, two types of messages need to be updated recursively in a factor graph, as shown in the Fig. 3.5, (1) messages from variable nodes to factor nodes; and (2) messages from factor nodes to variable nodes.

For the first case, since we have three different kinds of variable nodes in the graph, let's discuss them one by one. Starting from the response nodes, there are two types of outgoing messages, $m_{y_{(i,j)} \to f_{(i,j)}}$ and $m_{y_{(i,j)} \to f_{(i,j),(k,l)}}$, which can be calculated directly using Eq. (3.23). Since the relationship between the localized reduced random variable $\mathbf{s}_{(i,j)}$ and ξ is deterministic, there is no need to find the message sending from ξ to $\mathbf{s}_{(i,j)}$, and reverse versa. In other words, we don't need to calculate the message from $\mathbf{s}_{(i,j)}$ to $f_{\mathbf{s}_{(i,j)},\xi}$, and the message from ξ to $f_{\mathbf{s}_{(i,j)},\xi}$. However, to update the message from $\mathbf{s}_{(i,j)}$ to $f_{(i,j),(k,l)}$ and the message from $\mathbf{s}_{(i,j)}$ to $f_{(i,j)}$, we need to know the message from $f_{\mathbf{s}_{(i,j)},\xi}$ to $\mathbf{s}_{(i,j)}$ from the definition. This message is hard to obtain because both ξ to $\mathbf{s}_{(i,j)}$ are high dimensional, and in addition we need to know the messages from all the factor nodes connected to

ξ except $m_{f_{\mathbf{s}_{(i,j)},\xi}\to\xi}$. To bypass these difficulties, we use a different way to construct the unknown message from the information that is already known to us as follows. According to Eq. (3.25), we can write the following:

$$p(\mathbf{s}_{(i,j)}) \propto m_{f_{\mathbf{s}_{(i,j)},\xi}\to\mathbf{s}_{(i,j)}}(\mathbf{s}_{(i,j)}) \prod_{f\in\Gamma(\mathbf{s}_{(i,j)})\setminus f_{\mathbf{s}_{(i,j)},\xi}} m_{f\to\mathbf{s}_{(i,j)}}(\mathbf{s}_{(i,j)}). \tag{3.26}$$

Since $\mathbf{S}$ is known, we also know $p(\mathbf{s}_{(i,j)})$, and the message coming from the cross factor node $f_{(i,j)}$ and the message coming from the correlation factor node $f_{(i,j),(k,l)}$ can be computed from the previous iteration, where $y_{(k,l)} \in \Gamma(y_{(i,j)})$. Then we can write:

$$m^{(t)}_{f_{\mathbf{s}_{(i,j)},\xi}\to\mathbf{s}_{(i,j)}}(\mathbf{s}_{(i,j)}) \propto \frac{p(\mathbf{s}_{(i,j)})}{\prod_{f\in\Gamma(\mathbf{s}_{(i,j)})\setminus f_{\mathbf{s}_{(i,j)},\xi}} m^{(t-1)}_{f\to\mathbf{s}_{(i,j)}}(\mathbf{s}_{(i,j)})}. \tag{3.27}$$

As a result, the message sent from $f_{\mathbf{s}_{(i,j)},\xi}$ to $\mathbf{s}_{(i,j)}$ is updated using the known marginal distribution of $\mathbf{s}_{(i,j)}$ and $m^{(t-1)}_{f_{(i,j)}\to\mathbf{s}_{(i,j)}}(\mathbf{s}_{(i,j)})$ and $m^{(t-1)}_{f_{(i,j),(k,l)}\to\mathbf{s}_{(i,j)}}(\mathbf{s}_{(i,j)})$. In this work, this message is calculated by sampling from Eq. (3.27) using the Metropolis Hastings algorithm [45]. The details of how to calculate $m^{(t)}_{f_{\mathbf{s}_{(i,j)},\xi}\to\mathbf{s}_{(i,j)}}(\mathbf{s}_{(i,j)})$ are given in B.1.

For the second case, to update the messages from the factor nodes to the variable nodes, except the one from $f_{\mathbf{s}_{(i,j)},\xi}$ to $\mathbf{s}_{(i,j)}$, which is discussed above, and the one from $f_{\mathbf{s}_{(i,j)},\xi}$ to ξ, which we don't really care, all the others can be calculated via Eq. (3.24).

Remark 5**:** If a realization of the stochastic input, $\mathbf{a}$, is given, then all the message update is going to be held among the response nodes because $\mathbf{S}$ and ξ can be exactly known from the model reduction scheme. All the message update involving $\mathbf{s}_{(i,j)}$ terms is going to be replaced by $\delta_{\mathbf{s}_{(i,j)}}(\mathbf{s}^{(n)}_{(i,j)})$, where the Delta function only takes value when $\mathbf{s}^{(n)}_{(i,j)}$ equals to the given realization. After the belief

propagation algorithm is completed, we obtain the marginal distribution of the physical responses conditioned on the given input, e.g. $p(y_{(i,j)}|\mathbf{a})$. Let the expectation $\mathbb{E}[y_{(i,j)}|\mathbf{a}]$ be the predicted values of the physical responses and use the variance to measure the confidence about the mean prediction. We thus obtain a surrogate model based on the graphical model that for an any input realization provides us the response of the system as well as our confidence on this prediction.

In the uncertainty quantification problem, one exerts a known distribution on the input $\mathbf{A}$, $p(\mathbf{A})$, and is interested to quantify the probability induced by it on the response. Using the model reduction techniques discussed in Section 3.2, we can explicitly compute the distribution of ξ and $\mathbf{S}$, $p(\xi)$ and $p(\mathbf{S})$, respectively, given $p(\mathbf{A})$ or a set of realizations of $\mathbf{A}$. Then, by executing the inference problem discussed above, we can obtain an explicit representation of the marginal distribution of all the responses by gathering all the messages coming from the neighbors (as in Eq. (3.25)). The statistics of interest can be calculated directly from the posterior marginal distribution.

Nonparametric belief propagation

In the nonparametric graphical model, each message is represented by a Gaussian mixture. Then the belief update Eq. (3.24) becomes analytically intractable. Currently there are two possible approximations for performing the belief update. The first one is using a variational method [44]. The basic idea of the variational method is using a much simpler form (user defined) to obtain an approximation that is as close as possible to the target message. The second approach is using a sampling method [88]. The idea of the sampling method comes from

Algorithm 3: The complete inference algorithm

1: Initialization: With given $p(\Xi)$ and the deterministic relationship between $\mathbf{S}$ and ξ, $p(\mathbf{S})$ can be obtained via MC method as discussed in Section 3.2. We set the initial message as $m_{\xi}^{(0)}(\mathbf{s}_{(i,j)}) = p(\mathbf{s}_{(i,j)})$, and all other messages as a standard Gaussian $\mathcal{N}(0,1)$.

2: Iterate: At step t,

   1. Update message from variable nodes to factor nodes as in Eq. (3.23).
   2. Update message from factor nodes to variable nodes as in Eq. (3.24).
   3. Update $m_{f_{\mathbf{s}_{(i,j)},\xi}}^{(t)}$ as in Eq. (3.27).

3: Convergence: the algorithm stops when,

$$\epsilon = \frac{1}{N_{\mathcal{V}^{(y)}}} \sum_{y_{(i,j)} \in \mathcal{V}^{(y)}} \|\mu^{(t)}(y_{(i,j)}) - \mu^{(t-1)}(y_{(i,j)})\|_2^2 < \delta, \tag{3.28}$$

   where $\mu^{(t)}(y_{(i,j)})$ denotes the estimated mean of posterior marginal distribution of $y_{(i,j)}$ at step t.

particle filters. In [93], it was extended to graphs containing continuous, non-Gaussian variables leading to the so called "Nonparametric belief propagation (NBP) method". In this work, we utilize the NBP algorithm to perform the inference problem. Specifically, we use the NBP algorithm to approximately find the update of Eq. (3.24). In the following, we clearly demonstrate how to apply the NBP algorithm into our framework. The NBP algorithm approximates the belief update Eq. (3.24) using a sampling method [93]. It circumvents sampling directly from Eq. (3.24) (which is rather difficult task) by decomposing the process into two steps. For math convenience, denote $y_{-q} = \mathcal{Y}_f \setminus y_q$. In the first step, we draw N independent samples $\widetilde{y}_{-q}^{(n)}$ from a partial belief estimate combining

the marginal influence function of the potential function $f(\mathbf{y}_f)$ on y_{-q} and all the other incoming messages to the factor node f. The marginal influence function $\zeta(y_{-q})$ is defined by

$$\zeta(y_{-q}) = \int f(\mathbf{y}_f) dy_q. \tag{3.29}$$

In this work, $f(\mathbf{y}_f)$ is a Gaussian mixture, so $\zeta(y_{-q})$ is simply the Gaussian mixture obtained by marginalizing each component.

In the second step, for each of these auxiliary particles $\widetilde{y}_{-q}^{(n)}$, we make samples $\widetilde{y}_q^{(n)}$ from the normalized conditional potentials proportional to $f(y_q, y_{-q} = \widetilde{y}_{-q}^{(n)})$. The detailed algorithm of NBP is summarized in Algorithm 4.

Remark 6**:** Since we are using a Gaussian mixture to represent all messages, an inevitable problem will arise with the increase of the number of mixture components. For example, assume that all the messages are M-component Gaussian mixtures, and the BP belief update of Eq. (3.24) is defined by a product of h mixtures. The product of h Gaussian mixtures, each containing M components, is itself a mixture of M^h Gaussian distributions. While in principle this belief update could be performed exactly, the exponential growth in the number of mixture components quickly becomes intractable. Therefore, in this work, we use Gaussian mixture reduction via clustering (GMRC) method to reduce the number of mixture components whenever the number of mixture components of the beliefs exceed M. The details of GMRC algorithm are given in B.2.

Algorithm 4: The detailed algorithm for nonparametric belief propagation

Require: Input message $m^{(t)}_{y_r \to f}(y_r) = \{\omega^{(m)}_{y_r \to f}, \mu^{(m)}_{y_r \to f}, \Sigma^{(m)}_{y_r \to f}\}_{m=1}^{M}$ for each $y_r \in \mathcal{Y}_f \setminus y_q$.

Ensure: Construct an output message $m^{(t)}_{f \to y_q}(y_q)$.

1: Determine the marginal influence $\zeta(y_{-q})$ by Eq. (3.29).

2: Draw N independent, weighted samples from the product,

$$\widetilde{y}^{(n)}_{-q} \sim \zeta(y_{-q}) \prod_{y_r \in \mathcal{Y}_f \setminus y_q} m^{(t)}_{y_r \to f}(y_r). \tag{3.30}$$

GMRC (A Gaussian mixture reduction technique discussed in B.2) is first adopted to reduce the components of the product and then exact sampling method [53] is applied.

3: For each $\widetilde{y}^{(n)}_{-q}$, sample from,

$$\widetilde{y}^{(n)}_{q} \sim f(y_q, y_{-q} = \widetilde{y}^{(n)}_{-q}). \tag{3.31}$$

4: Construct $m^{(t)}_{f \to y_q}(y_q)$ from $\widetilde{y}^{(n)}_{q}$ by taking $\widetilde{y}^{(n)}_{q}$ as realizations of message $m^{(t)}_{f \to y_q}(y_q)$. Specifically, assume $m^{(t)}_{f \to y_q}(y_q) \propto \sum_{m=1}^{M} \omega^{(m)}_{f \to y_q} \mathcal{N}(\mu^{(m)}_{f \to y_q}, \Sigma^{(m)}_{f \to y_q})$, where the unknowns are $\{\omega^{(m)}_{f \to y_q}, \mu^{(m)}_{f \to y_q}, \Sigma^{(m)}_{f \to y_q}\}_{m=1}^{M}$. These can be learned using the EM algorithm by taking $\widetilde{y}^{(n)}_{q}$ as training samples.

3.5 Numerical examples

In this paper, we construct a probabilistic graphical model to study two-dimensional, single phase, steady-state fluid flow through random heterogeneous porous media. A review of the mathematical models of flow through porous media can be found in [2]. The spatial domain D is chosen to be the unit square $[0, 1]^2$, representing an idealized oil reservoir. Let us denote with p and $\mathbf{u}$ the pressure and the velocity fields of the fluid, respectively. These are

connected via the Darcy law:

$$\mathbf{u} = -\mathbf{K}\nabla p, \text{in } D, \tag{3.32}$$

where $\mathbf{K}$ is the permeability tensor that models the easiness with which the liquid flows through the reservoir. Combining the Darcy law with the continuity equation, it is easy to show that the governing PDE for the pressure is:

$$-\nabla \cdot (\mathbf{K}\nabla p) = q, \text{in } D, \tag{3.33}$$

where the source term q may be used to model injection/production wells. In this example, we consider square wells: an injection well on the left-bottom corner of D and a production well on the top-right corner. The particular mathematical form of the source term q is as follows:

$$q(\mathbf{x}) = \begin{cases} -r, & \text{if } |x_i - \frac{1}{2}w| < \frac{1}{2}w, \text{ for } i = 1, 2, \\ r, & \text{if } |x_i - 1 + \frac{1}{2}w| < \frac{1}{2}w, \text{ for } i = 1, 2, \\ 0, & \text{otherwise,} \end{cases} \tag{3.34}$$

where r specifies the rate of the wells, w their size (chosen here to be $r = 10$ and $w = 1/8$), and $\mathbf{x} = (x_1, x_2) \in D$. Furthermore, we impose no-flux boundary conditions on the walls of the reservoir:

$$\mathbf{u} \cdot \tilde{\mathbf{n}} = 0, \text{on } \partial D, \tag{3.35}$$

where $\tilde{\mathbf{n}}$ is the unit normal vector to the boundary. These boundary conditions specify the pressure p up to an additive constant. To assure uniqueness of the boundary value problem defined by Eqs. (3.32), (3.33) and (3.35), we impose the constraint [16]:

$$\int_D p(\mathbf{x}) d\mathbf{x} = 0. \tag{3.36}$$

The boundary value problem is solved using a mixed finite element formulation. We use first-order Raviart-Thomas elements for the velocity [79], and

zero-order discontinuous elements for the pressure [19]. The permeability is defined on a 64×64 fine-grid and we are interested in the physical responses on a 8×8 coarse-grid. The solver was implemented using the Dolfin C++ library [63]. The eigenfunctions of the exponential random field used to model the permeability were calculated via Stokhos which is part of Trilinos [49].

In this work, the final responses taken into consideration include $x-$velocity, u_x, $y-$velocity, u_y and pressure, p. We assume independence of the multiple output responses so we can build an independent graphical model for each of them. This is typical of many uncertainty quantification methods but methodologies where correlations are accounted can be considered as well [108]. As previously discussed, the constructed graphical model is a nonparametric model. Then naturally an important question arises as to what the proper number is of the mixture components considered. One should avoid to choose a large number due to the exponential increase for the computational cost, especially at the loop belief propagation step. A large number of mixture components does not necessarily lead to better results and for some cases may lead to over-fitting. In the context of the examples presented below, three mixture components were shown to provide an adequate choice for the accuracy desired.

3.5.1 Stationary random field

In this example, the log-permeability is considered as a stationary random field. We restrict ourselves to an isotropic permeability tensor:

$$K_{ij} = K\delta_{ij}. \tag{3.37}$$

K is modeled as

$$K(\mathbf{x}) = \exp\{G(\mathbf{x})\}, \tag{3.38}$$

where G is a Gaussian random field:

$$G(\cdot) \sim \mathcal{N}(m, c_G(\cdot,\cdot)), \tag{3.39}$$

with constant mean m and an exponential covariance function given by

$$c_G(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = v_G^2 \exp\left\{-\frac{|x_1^{(1)} - x_1^{(2)}|}{l} - \frac{|x_2^{(1)} - x_2^{(2)}|}{l}\right\}. \tag{3.40}$$

The parameter l represents the correlation lengths of the field, while $v_G > 0$ is its variance. In order to obtain a finite dimensional representation of G, we employ the Karhunen-Loève expansion [39] and truncate it after k_ξ terms:

$$\boldsymbol{G(\xi; \mathbf{x}) = m + \sum_{k=1}^{k_\xi} \xi_k \phi_k(\mathbf{x}),} \tag{3.41}$$

where $\xi = (\xi_1, \ldots, \xi_{k_\xi})$ is a vector of independent, zero mean and unit variance Gaussian random variables and $\phi_k(\mathbf{x})$ are the eigenfunctions of the exponential covariance given in Eq. (3.40) (suitably normalized).

The values we choose for the parameters are $m = 0, l = 0.1$ and $v_G = 1$ in Eq. (3.40), and $k_\xi = 50$ in Eq. (3.41). In the following, we first verify the model reduction framework in Section 3.5.1 and then we move to the inference tasks on the graph: 1) given the input distribution of ξ, investigate how the uncertainty propagates to the response in Section 3.5.1; 2) given a new permeability field, find the prediction of unobserved responses with proper error bars in Section 3.5.1.

Model reduction

As discussed in Section 3.2, PCA model reduction technique is applied to reduce the dimensionality of the input permeability field. Fig. 3.6 shows the normalized eigen-plot and energy-plot for the PCA reduction for the input permeability over the corresponding coarse-elements to two random coarse-nodes. Here, "normalized" means that each eigenvalue is divided by the sum of all the eigenvalues. As shown on these plots, by using less than ten eigenvectors, the cumulative preserved energy is almost one, which means a ten-dimensional random variable representation is enough to describe the original data set.

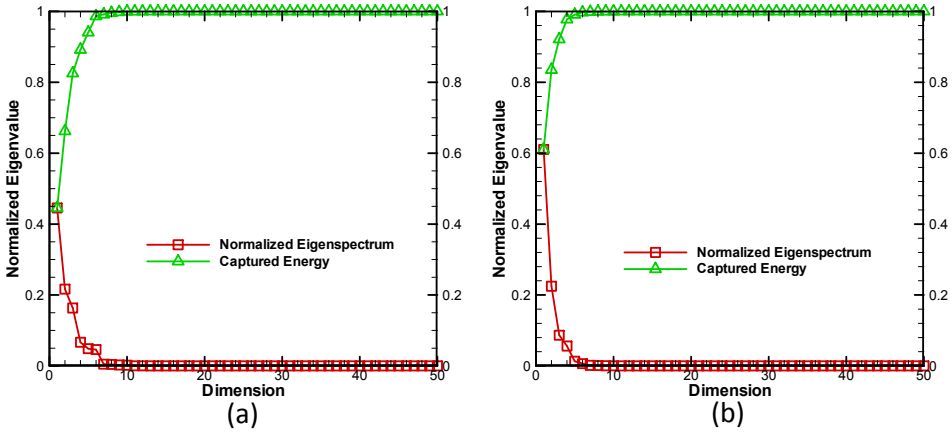

Figure 3.6: Stationary random field: Normalized eigenspectrum and energy plot for the input permeability in two random subdomains.

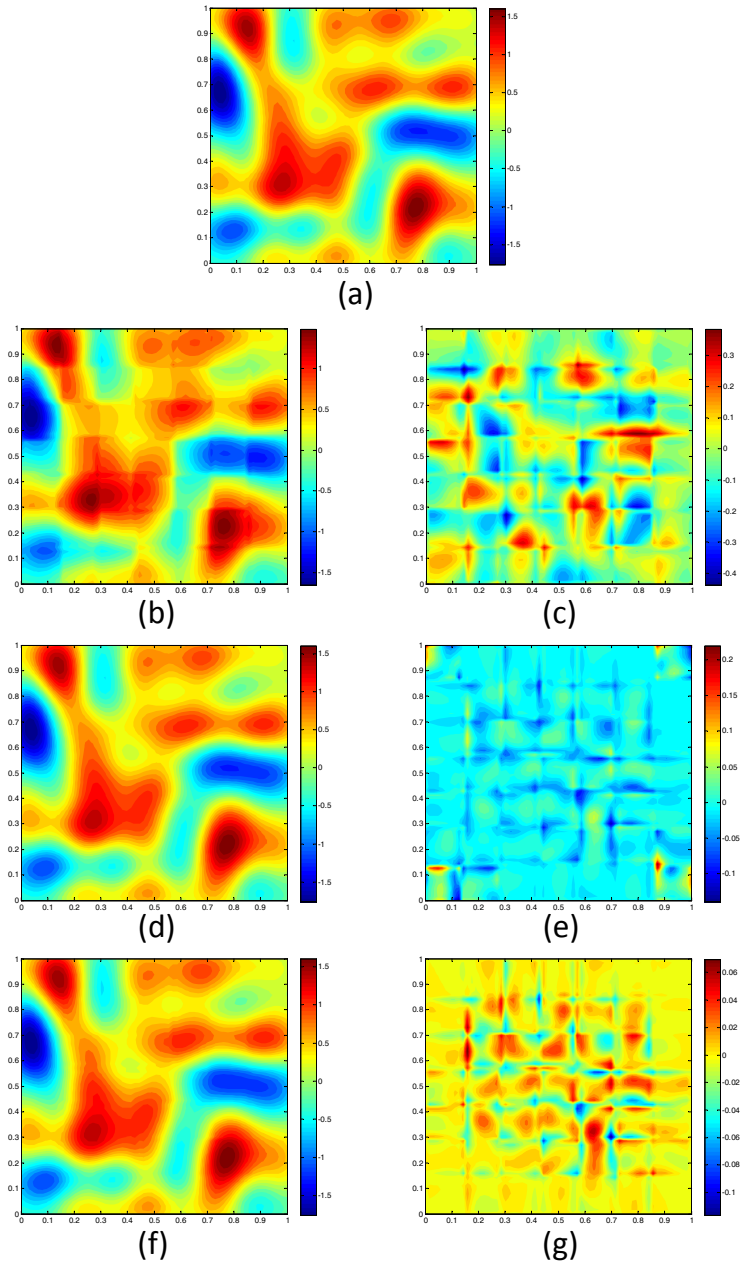

Figure 3.7: Stationary random field: Comparison of the reconstructed input permeability field with the original given sample. (a) with different number of training data for $k = 10$, where k is the dimensionality of the reduced space; (b)(d)(f) The reconstructed input permeability using $N = 200, 1000$, and 4000 training data, respectively; (c)(e)(g) The error between the reconstructed permeability field and the original sample.

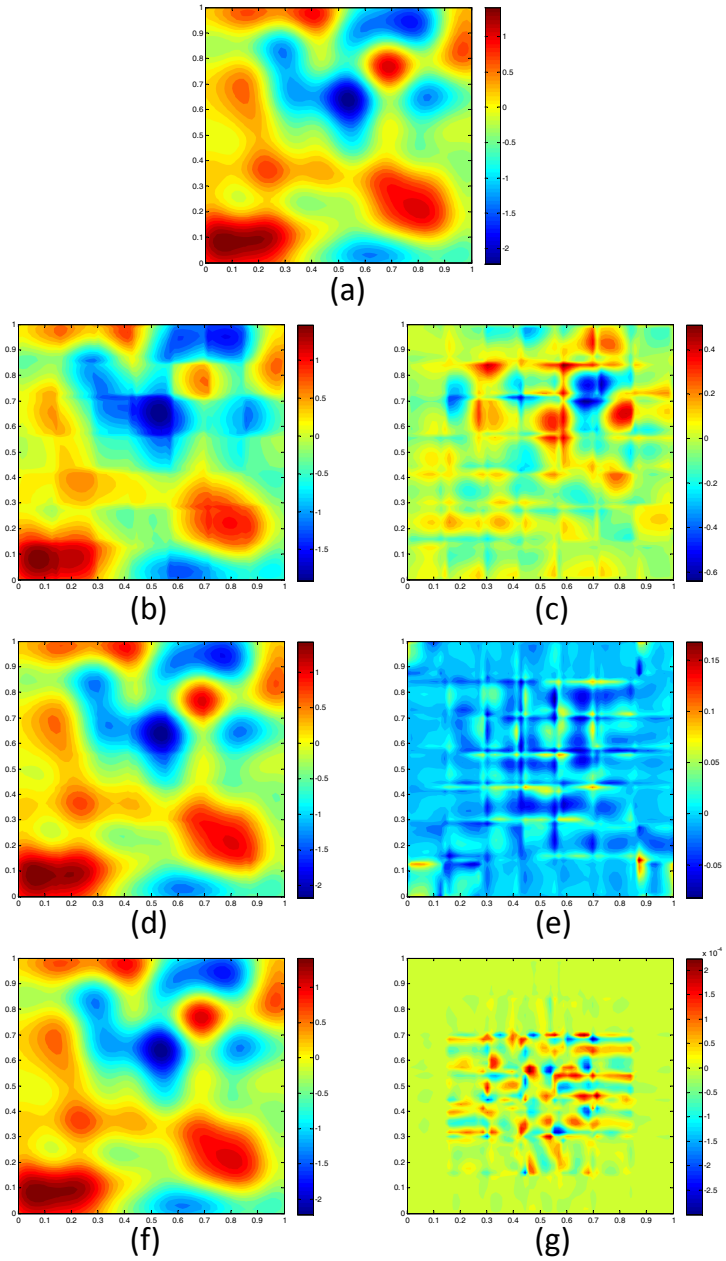

Figure 3.8: Stationary random field: Comparison of the reconstructed input permeability field with the original given sample. (a) with different k for $N = 1000$; (b)(d)(f) The reconstructed input permeability using $k = 5, 10$, and 30, respectively; (c)(e)(g) The error between the reconstructed permeability field and the original sample.

Then, we compare the reconstruction error of the input permeability field

with different number of training data and different reduced dimensionality k (Fig. 3.9). The reconstruction error is computed by

$$e = \frac{1}{N_g N} \sum_{n=1}^{N} \|\mathbf{A}^{(n)} - \widetilde{\mathbf{A}}^{(n)}\|_2^2, \tag{3.42}$$

where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{N_f}]$, N_f is the number of elements on the fine-mesh and $\widetilde{\mathbf{A}} = [\widetilde{\mathbf{a}}_1, \widetilde{\mathbf{a}}_2, \ldots, \widetilde{\mathbf{a}}_{N_f}]$ where $\widetilde{\mathbf{a}}_i$ is the reconstructed permeability on the fine-element e_i. The superscript (n) denotes the n-th sample and N is the total number of samples used. The given figure indicates that the number of reduced dimensionality k has a higher impact on the final performance of the reconstruction than the number of training data. The reduced dimensionality k is chosen as 10 in this problem.

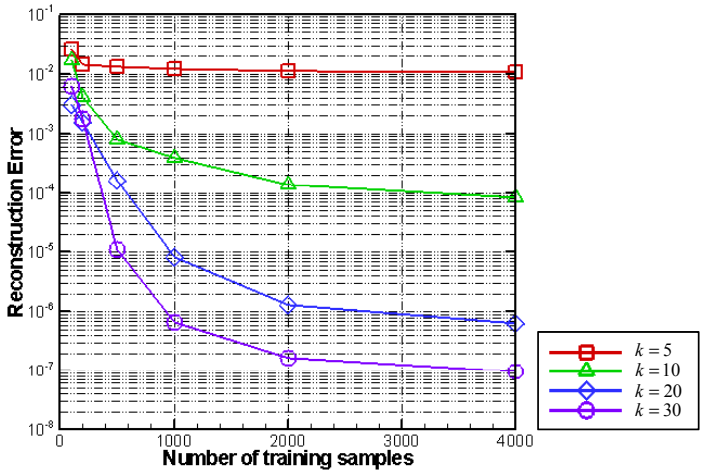

Figure 3.9: Stationary random field: Comparison of the reconstruction error of the input permeability field with different number of training data, and different reduced dimensionality k.

Uncertainty propagation

In this section, we are going to investigate how the uncertainties propagate from the input permeability to the output (velocity and pressure) response. After the

graphical model is completely learnt by the training data, we send the distribution of $\mathbf{S}$, $p(\mathbf{S})$, which is calculated from the known input distribution $p(\xi)$, to the graphical model as the input message, and then run the nonparametric belief propagation algorithm. After all the messages in the graph converge, we compute the conditional marginal distribution of the response variables by combining all the messages coming into the response variable as in Eq. (3.25).

Fig. 3.10 compares the predicted mean of u_x with a Monte Carlo estimate using 10^5 observations. We can clearly see that as the number of training data increases, the prediction gets more and more accurate. The same statistic for u_y and p is reported in Figs. 3.11 and 3.12, respectively.

Fig. 3.13 compares the predicted variance of u_x to a Monte Carlo estimate using 10^5 observations. Also, the predicted variance converges to the MC results with the increase of the number of the training data. The same statistic for u_y and p is given in Figs. 3.14 and 3.15, respectively. We can see that $N = 400$ training samples can already give rather accurate predictions for the marginal mean and variance of the responses. Notice that in this work, the predicted marginal probability is given in a Gaussian mixture form with three components. For example, the response at one coarse-grid node, $y_{(i,j)}$ (random variable) can be represented as $y_{(i,j)} = \sum_{m=1}^{M} \omega_{(i,j)}^{(m)} y_{(i,j)}^{(m)}$, where $y_{(i,j)}^{(m)} \sim \mathcal{N}(\mu_{(i,j)}^{(m)}, (\sigma_{(i,j)}^{(m)})^2)$. The first-order and second-order statistics can be obtained by $\mathbb{E}[y_{(i,j)}] = \sum_{m=1}^{M} \omega_{(i,j)}^{(m)} \mu_{(i,j)}^{(m)}$ and $Var[y_{(i,j)}] = \sum_{m=1}^{M} (\omega_{(i,j)}^{(m)})^2 (\sigma_{(i,j)}^{(m)})^2$, respectively.

The error of the statistics is evaluated using the (normalized) L_2 norm of the error in variance defined by:

$$E_{L_2} = \sqrt{\frac{1}{N_G} \sum_{i=1}^{N_G} (v_{i,MC} - \widetilde{v}_i)^2}, \tag{3.43}$$

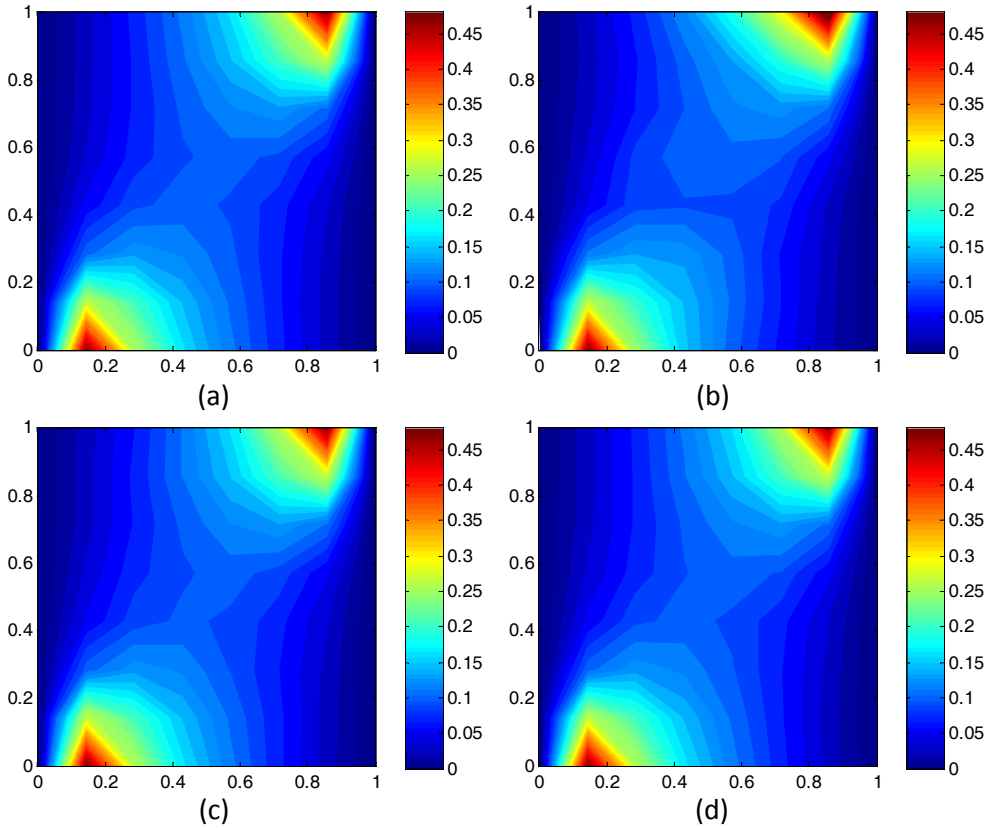

Figure 3.10: Stationary random field: Mean of u_x. (a) MC estimate using 10^5 observations; (b)(c)(d) The predicted mean of u_x using 50, 100, and 400 training samples, respectively.

where $v_{i,MC}$ is the Monte Carlo estimate of the variance of the response on the i-th coarse-node using 10^5 samples, and $\widetilde{v}_i$ is the predictive variance given by the graphical model. In Fig. 3.16, we plot the L_2 norm of the error as a function of the number of samples for u_x, u_y and p and a comparison with the MC results is shown. In addition, we compare the predicted probability densities of u_x, u_y and p at physical positions $(0.429, 0.429)$ and $(0.571, 0.571)$, with the PDFs obtained from the MC estimate using 10^5 observations, as shown in Figs. 3.17 and 3.18, respectively. From the figures, we can see that the PDFs do not have symmetric tails, so obviously, they are not Gaussian distributions. This is especially true for the velocity components that should be positive. As the number of observa-

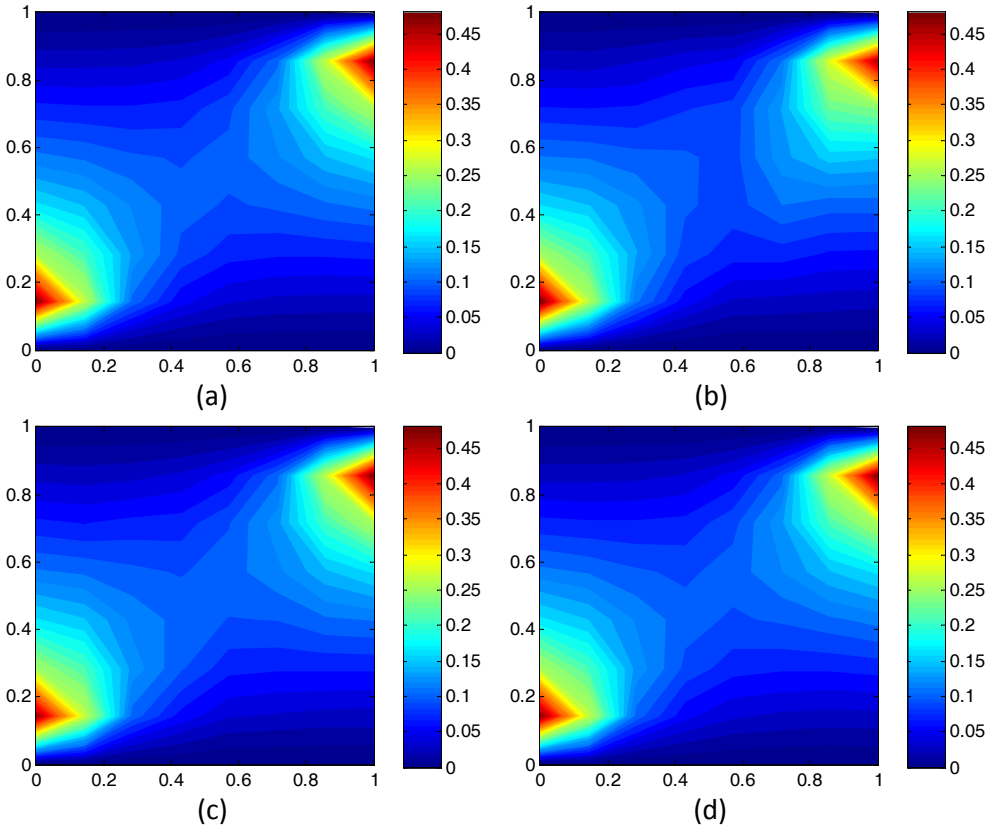

Figure 3.11: Stationary random field: Mean of u_y. (a) MC estimate using 10^5 observations; (b)(c)(d) The predicted mean of u_y using 50, 100, and 400 training samples, respectively.

tions increases, we can observe that the graphical model prediction gradually captures the major key features of the PDFs.

Response Prediction

In this section, we will show that the constructed graphical model is also capable of acting as a surrogate model of the deterministic solver. The problem can be described as follows: Given a new observation of the permeability field, $\mathbf{a}$, the objective is to obtain the conditional distribution $p(\mathbf{Y}|\mathbf{A} = \mathbf{a})$. With a new realization of the permeability field $\mathbf{a}$, we first compute the localized reduced in-

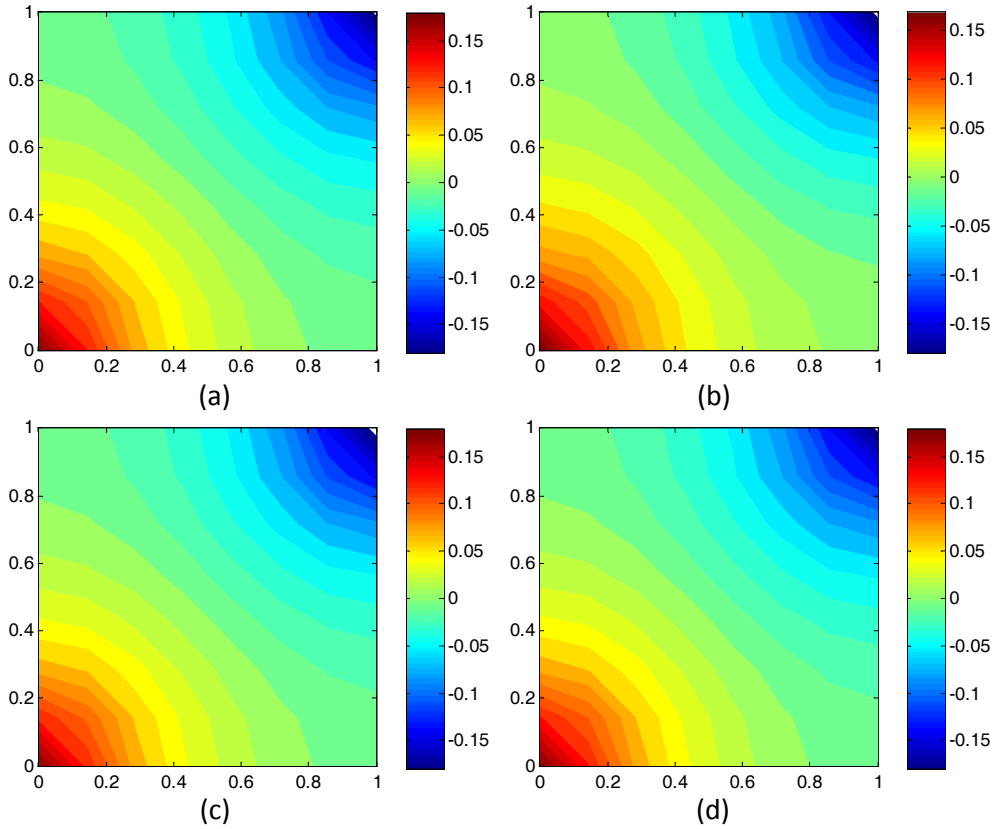

Figure 3.12: Stationary random field: Mean of p. (a) MC estimate using 10^5 observations; (b)(c)(d) The predicted mean of p using 50, 100, and 400 training samples, respectively.

put variables $\mathbf{S}(\mathbf{a})$. After that, instead of using the distribution of $\mathbf{S}$, $p(\mathbf{S})$, we use a Kronecker Delta function $\delta_{\mathbf{S}}(\mathbf{S}(\mathbf{a}))$ as the input message, and we send it to the pre-learned graphical model to execute the nonparametric belief propagation algorithm. Notice that now, all the potential functions involving the $\mathbf{S}$ variable need to be multiplied by the Kronecker Delta function $\delta_{\mathbf{S}}(\mathbf{S}(\mathbf{a}))$, as discussed in Section 3.4.2. Fig. 3.32 shows a comparison of the predicted u_x, u_y and p fields, with the results of the deterministic solver for given a new input permeability field $\mathbf{a}$, using $N = 400$ training samples. This permeability sample was generated from the same process as the training data. As shown from the figures, the predictions capture the main features of the responses.

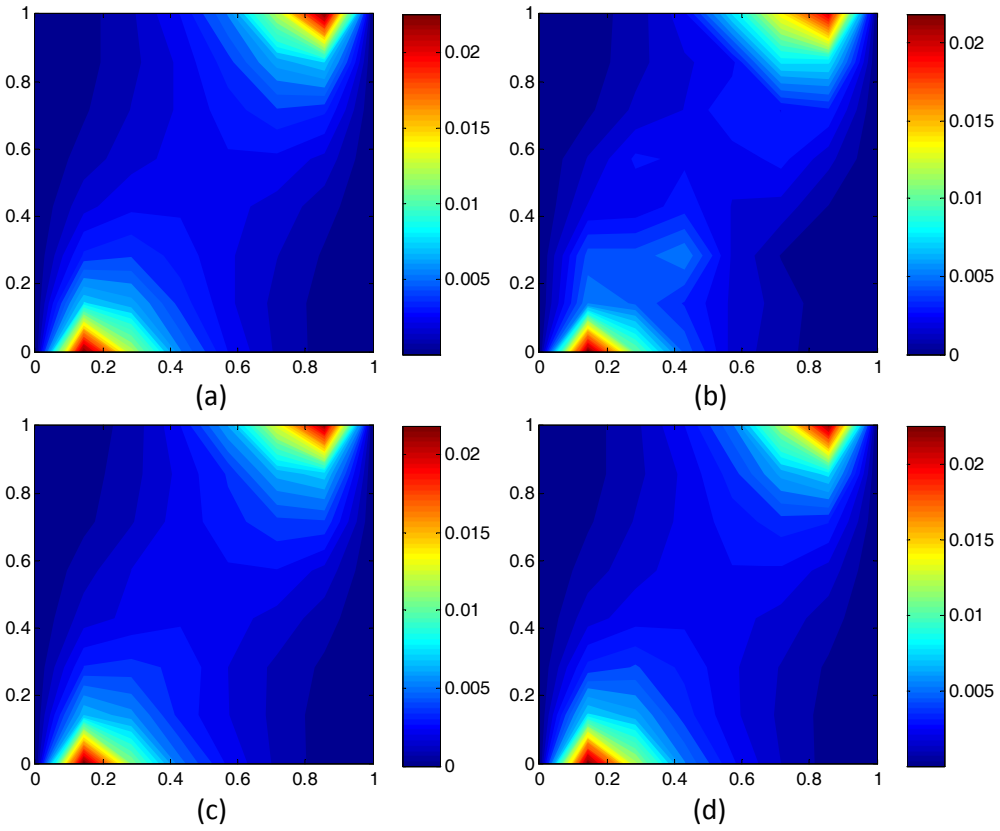

Figure 3.13: Stationary random field: Variance of u_x. (a) MC estimate using 10^5 observations; (b)(c)(d) The predicted variance of u_x using 50, 100, and 400 training samples, respectively.

3.5.2 Non-stationary random field

In the previous example, it was assumed that the porous media considered was stationary such that the covariance between any two points in the domain depends on their distance rather than their actual locations. However, hydraulic properties may exhibit spatial variations at various scales. Therefore, it is important to extend the probabilistic graphical model to non-stationary random fields. In this example, we use a non-stationary random field as stochastic input. The log-permeability on the k-th coarse-element is still a Gaussian random field

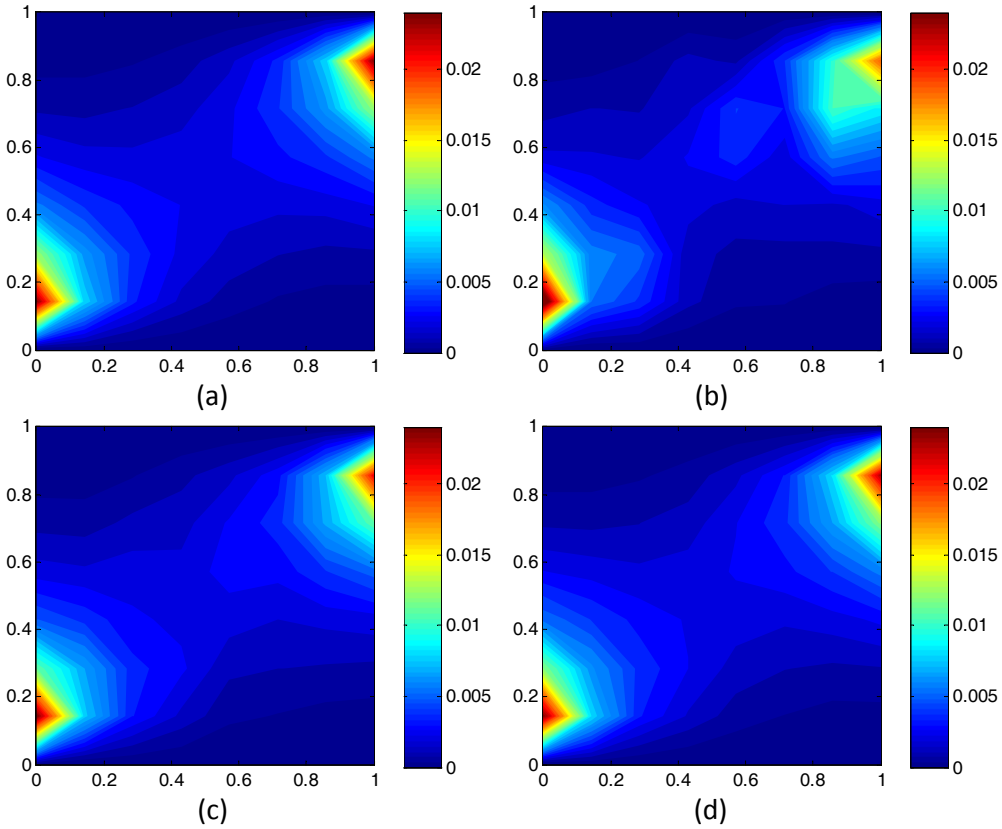

Figure 3.14: Stationary random field: Variance of u_y. (a) MC estimate using 10^5 observations; (b)(c)(d) The predicted variance of u_y using 50, 100, and 400 training samples, respectively.

with mean zero and an exponential covariance function, as given in Eq. (3.40):

$$c_G\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right) = v_G^2 \exp\left\{-\frac{|x_1^{(1)} - x_1^{(2)}|}{l_{k,1}} - \frac{|x_2^{(1)} - x_2^{(2)}|}{l_{k,2}}\right\}. \tag{3.44}$$

However, the correlation length in the non-stationary case is not a constant anymore. Since the coarse-grid has $N_x = 8$ rows and $N_y = 8$ columns of elements, we define the coordinate of the k-th element as (i_k, j_k) where i_k is the index in row and j_k is the index in column. Then the correlation length is set to be $l_{k,1} = 0.1 + \frac{0.4}{N_y - 1} j_k$ and $l_{k,2} = 0.1 + \frac{0.4}{N_x - 1} i_k$. The source term q is set to zero. Flow is induced from left to right side with Dirichlet boundary conditions $\bar{p} = 1$ on $x = 0$, $\bar{p} = 0$ on $y = 1$. No−flow Neumann boundary conditions are applied on the other two

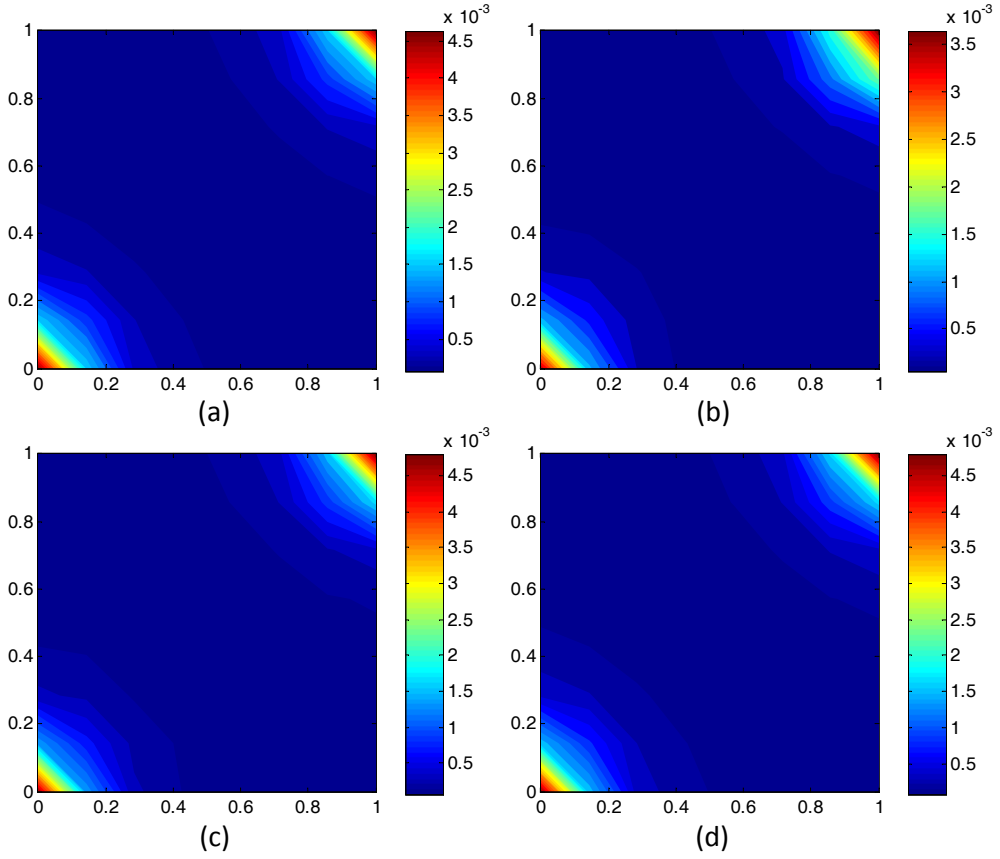

Figure 3.15: Stationary random field: Variance of p. (a) MC estimate using 10^5 observations; (b)(c)(d) The predicted variance of p using 50, 100, and 400 training samples, respectively.

sides of the square domain.

In this example, we investigate the non-stationary problem in a similar way as the previous stationary case. First, we verify the model reduction framework in Section 3.5.2 and then we investigate the uncertainty propagation from the inputs to the responses in Section 3.5.2. Finally, we use the graphical model to predict the responses given a new permeability field, in Section 3.5.2.

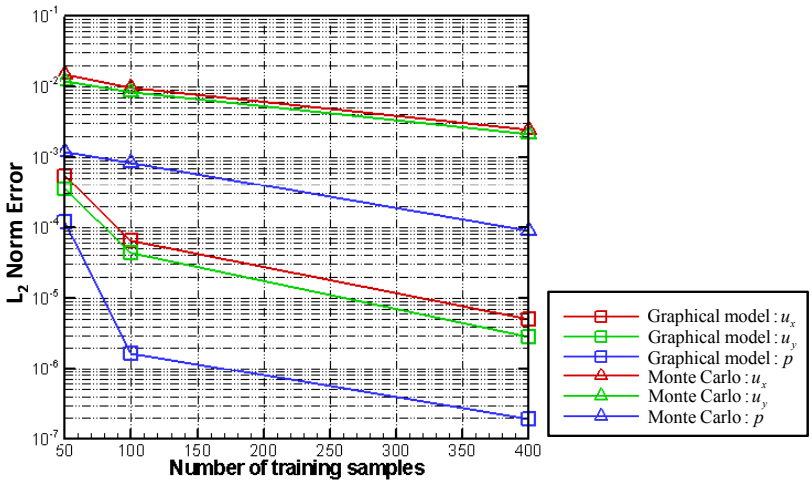

Figure 3.16: Stationary random field: The L_2 norm of the error as a function of the number of samples observed for graphical model framework.

Model reduction

In this section, we first compare the reconstructed input permeability field with the original one for different k, where k is the dimensionality of the reduced space, as shown in Fig. 3.20. Fig. 3.21 shows the comparison of the reconstructed input permeability field with the original given sample for different number of training samples N. In comparison with the reconstruction results in the previous example in Section 3.5.1, a higher k is needed here to obtain a relatively good reconstruction. This is expected because the non-stationary permeability field is much more complicated than the stationary case. We obtain the similar conclusions from these figures as in the earlier example. As the reduced dimensionality k increases, the reconstructed permeability field gets closer to the original permeability. Also as the number of training data N used increases, the reconstructed permeability becomes closer to the original realization.

We compare the reconstruction error of the input permeability field with dif-

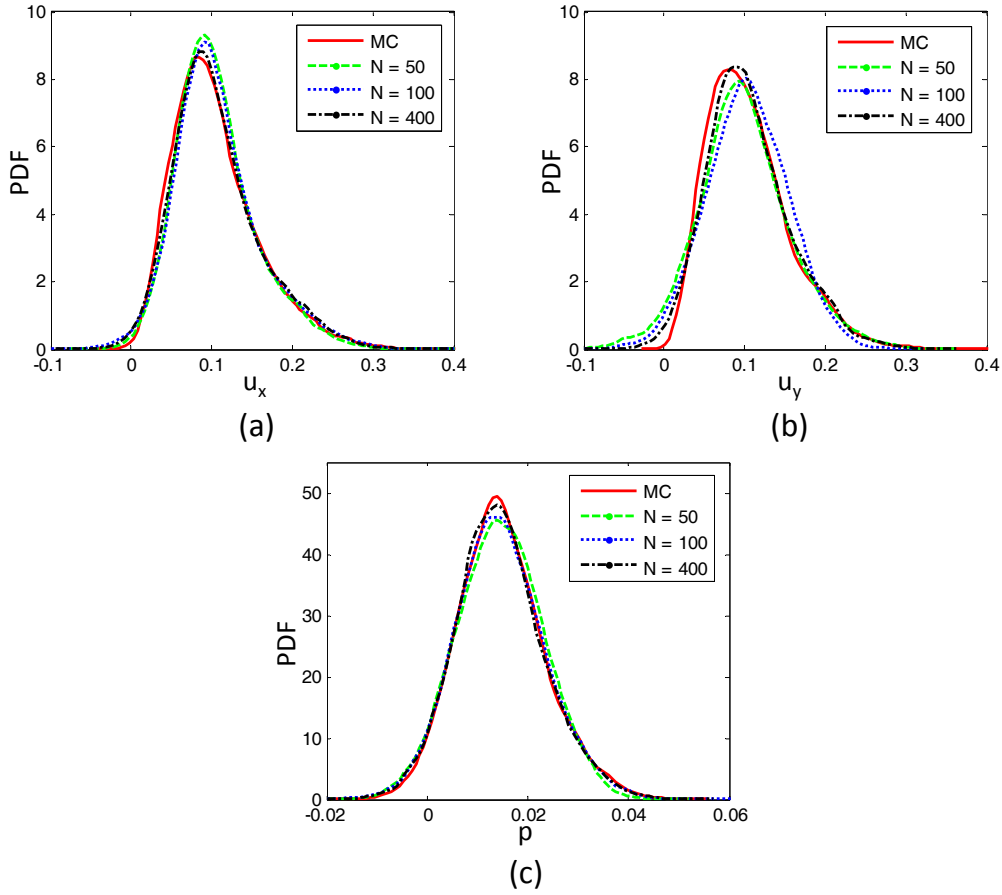

Figure 3.17: Stationary random field: Comparison of the predicted PDFs using different training data with the MC estimate at physical position $(0.429, 0.429)$ (a) u_x, (b) u_y, (c) p.

ferent number of training data N and different reduced dimensionality k using Eq. (3.42) in section 3.5.1, as shown in Fig. 3.22. In comparison with the reconstruction results in the previous example in Section 3.5.1, a higher k is needed here to obtain a relatively good reconstruction. This is expected because the non-stationary permeability field is much more complicated than the stationary case. In this example, k is chosen as 20, that is, the dimensionality of each $\mathbf{s}$ variable is 20.

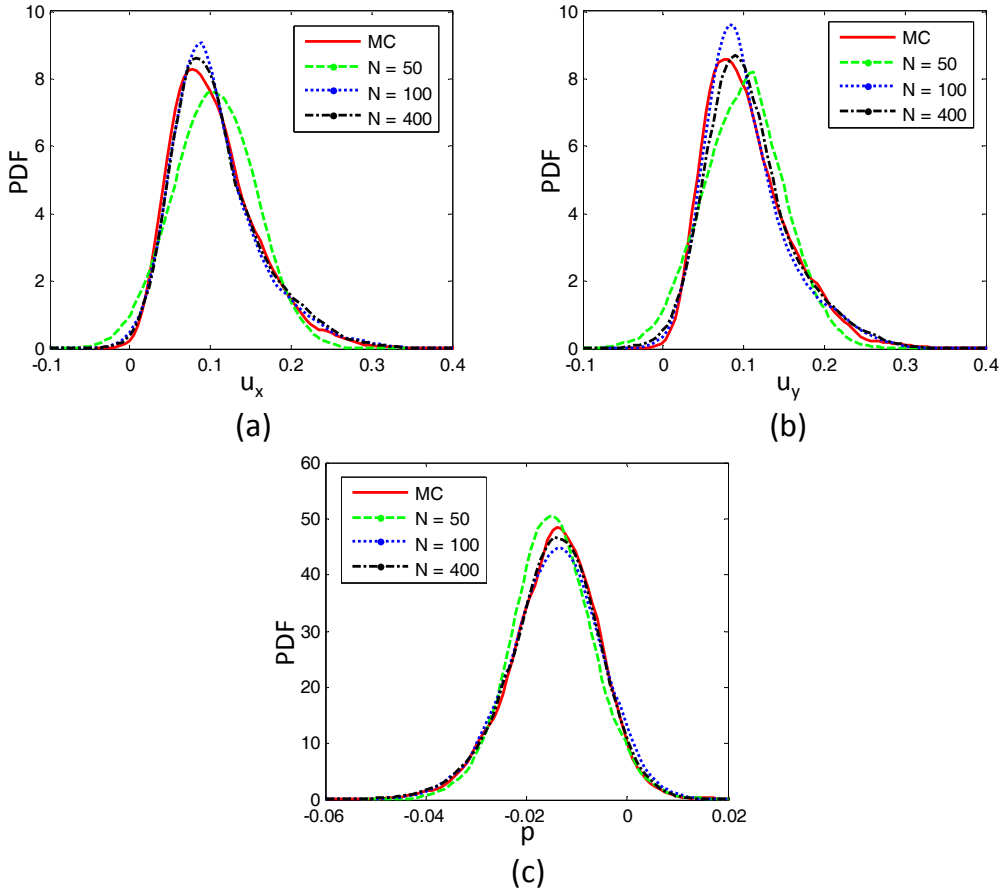

Figure 3.18: Stationary random field: Comparison of the predicted PDFs using different training data with the MC estimate at physical position $(0.571, 0.571)$ (a) u_x, (b) u_y, (c) p.

Uncertainty propagation

In this section, we are also going to investigate how the uncertainty propagate from the input permeability to the output (velocity and pressure) response, as in Section 3.5.1. Fig. 3.23 compares the predicted mean of u_x with a Monte Carlo estimate using 10^6 observations. We can clearly see that as the number of training data increases, the prediction gets more and more accurate. The same statistic for u_y and p is reported in Figs. 3.24 and 3.25, respectively.

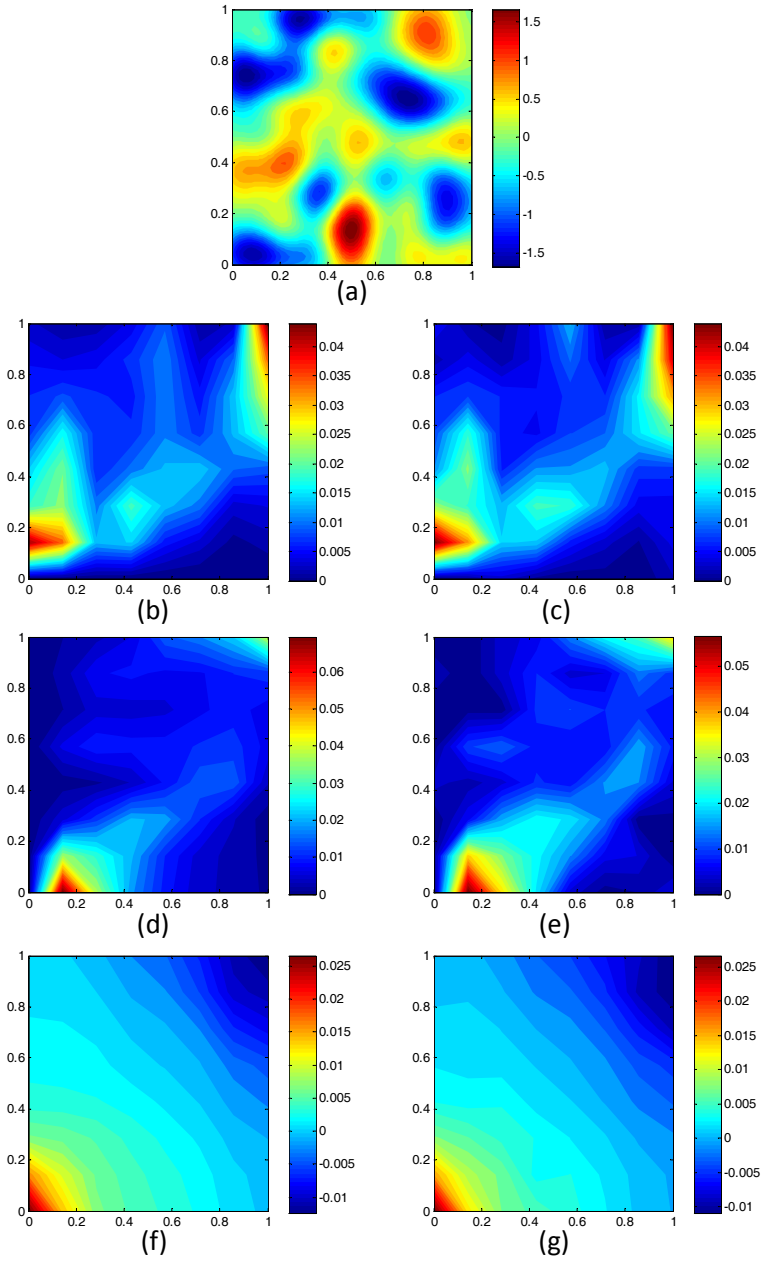

Figure 3.19: Stationary random field: Comparison of the predicted physical responses given a realization of stochastic input permeability with the true response. (a) The new observed input permeability field; (b)(d)(f) The true responses for the given permeability realization, from top to bottom, u_x, u_y and p, respectively; (c)(e)(g) The predicted means for u_x, u_y and p by graphical model using $N = 400$ training data, respectively.

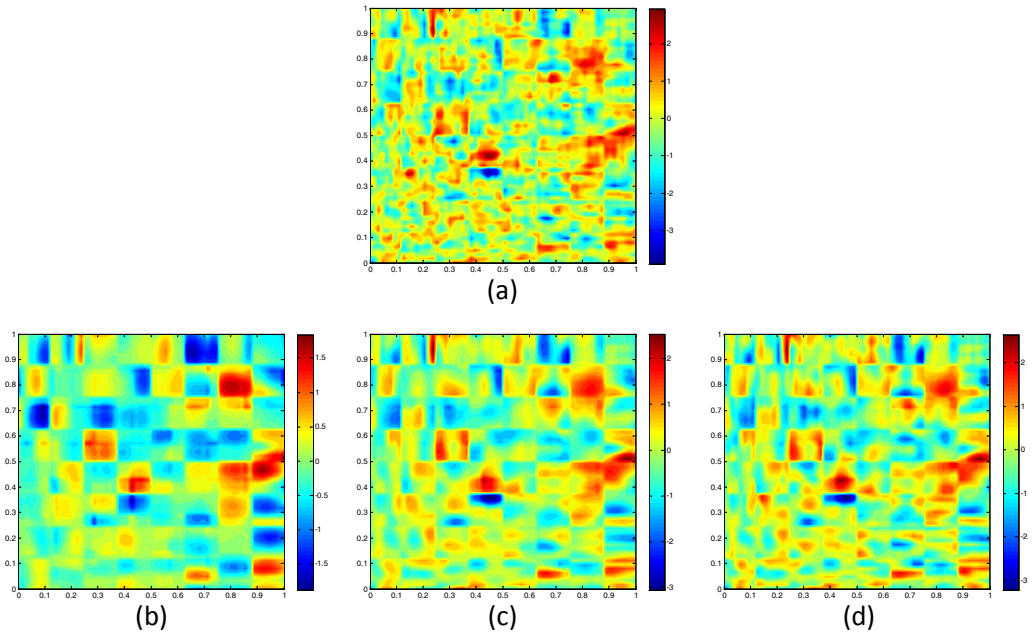

Figure 3.20: Non-stationary random field: Comparison of the reconstructed input permeability field with the original given sample (a) With different k for $N = 2000$; (b)(c)(d) The reconstructed input permeability using $k = 10, 30$, and 50, respectively.

Fig. 3.26 compares the predicted variance of u_x to a Monte Carlo estimate using 10^6 observations. Also, the predicted variance converges to the MC results with the increase of the number of the training data. The same statistic for u_y and p is given in Figs. 3.27 and 3.28, respectively. We can see that in this example, $N = 2000$ training samples can only provide a reasonable predictions for the marginal mean and variance of the responses, while in the previous stationary example in section 3.5.1, $N = 400$ training samples can already give rather accurate predictions. The predictions of the pressure, compared to the predictions of velocity, are much more accurate, this is because the pressure has less variability than the velocity in porous media flow problem.

Similarly, in Fig. 3.29, we plot the L_2 norm (Eq. (3.43)) of the error as a func-

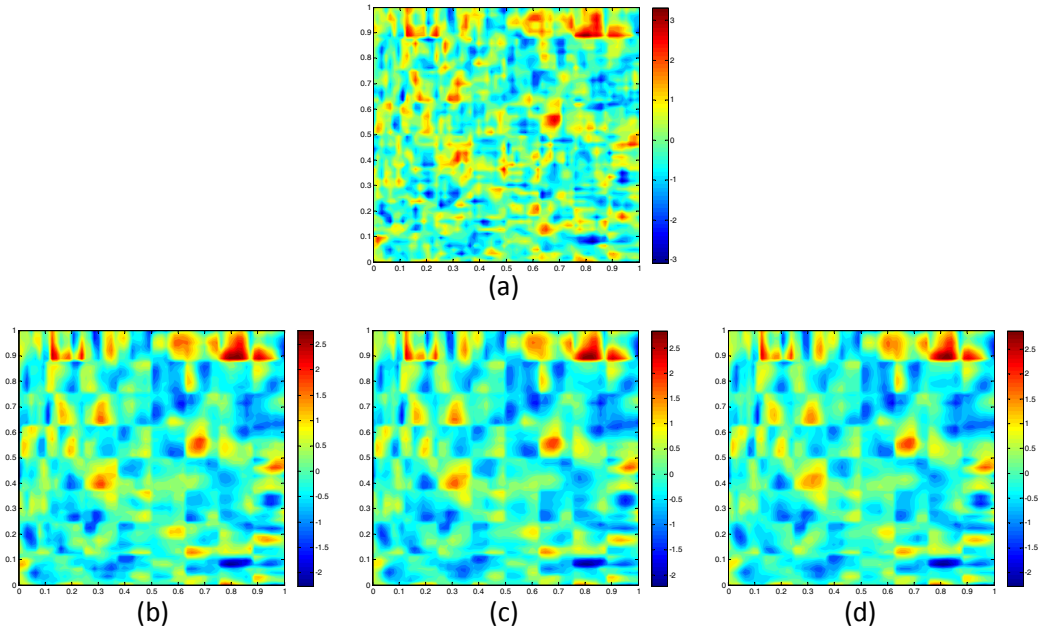

Figure 3.21: Non-stationary random field: Comparison of the reconstructed input permeability field with the original given sample (a) With different number of training data for $k = 30$, where k is the dimensionality of the reduced space; (b)(d)(f) The reconstructed input permeability using $N = 1000, 2000,$ and 4000 training data, respectively.

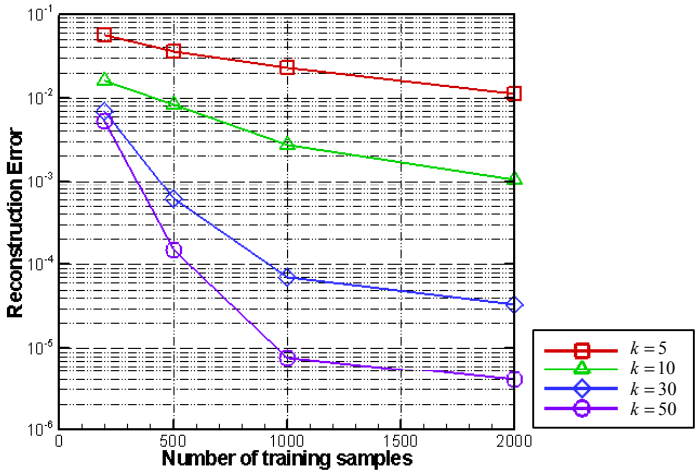

Figure 3.22: Non-stationary random field: Comparison of the reconstruction error of the input permeability field with different number of training data, and different reduced dimensionality k.

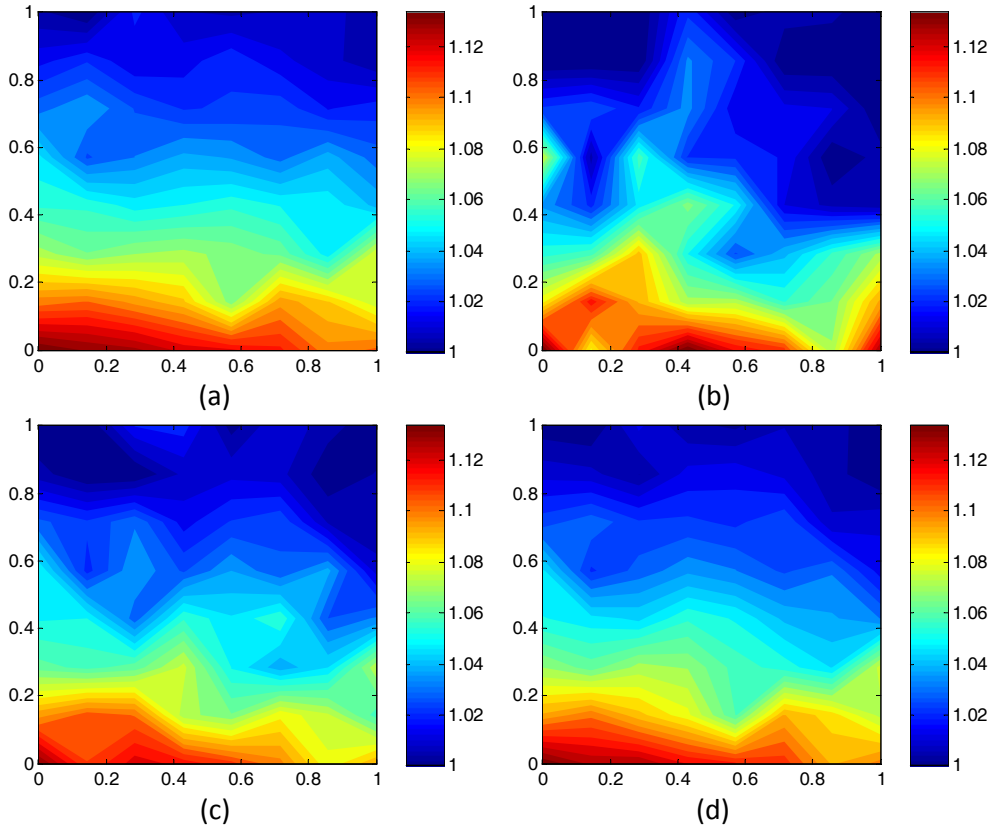

Figure 3.23: Non-stationary random field: Mean of u_x. (a) MC estimate using 10^6 observations; (b)(c)(d) The predicted mean of u_x using $200, 800$, and 2000 training samples, respectively.

tion of the number of samples for u_x, u_y and p and compare with the MC results. In addition, we compare the predicted probability densities of u_x, u_y and p at physical positions $(0.429, 0.429)$ and $(0.571, 0.571)$, with the PDFs obtained from the MC estimate using 10^6 observations, as shown in Figs. 3.30 and Fig. 3.31, respectively. From the figures, we can see that as the number of observations increases, the graphical model prediction gradually captures the major key features of the PDFs.

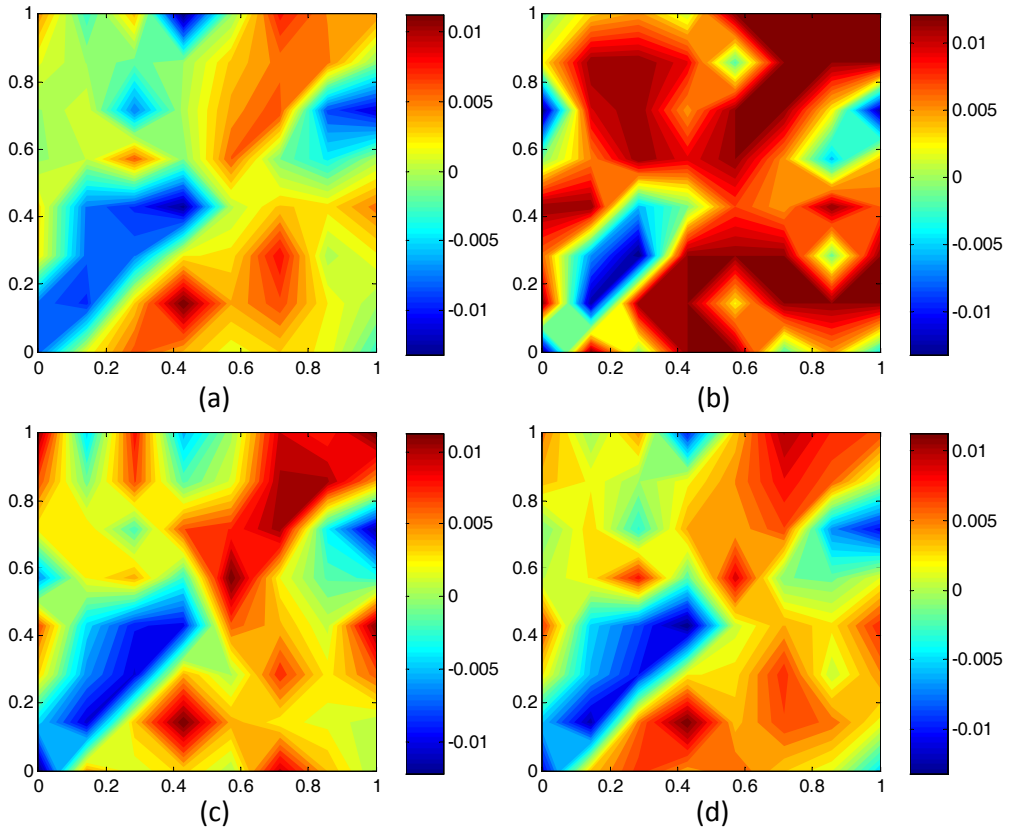

Figure 3.24: Non-stationary random field: Mean of u_y. (a) MC estimate using 10^6 observations; (b)(c)(d) The predicted mean of u_y using $200, 800$, and 2000 training samples, respectively.

Response Prediction

In this section, we also provide an example to demonstrate that the constructed graphical model is capable of acting as a surrogate model for the deterministic solver, as in section 3.5.1. Fig. 3.32 shows a comparison of the predicted u_x, u_y and p fields, with the results of the deterministic solver for given a new input permeability field $\mathbf{a}$, using $N = 2000$ training samples. As shown from the figures, the predictions capture the main features of the responses.

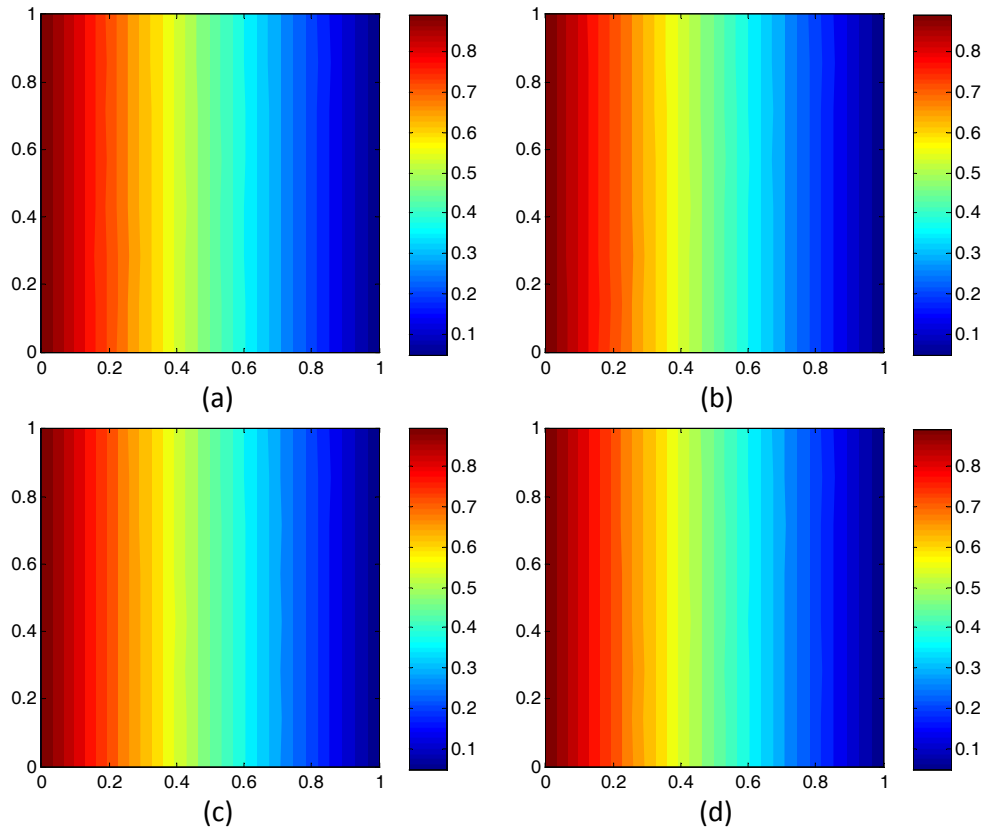

Figure 3.25: Non-stationary random field: Mean of p. (a) MC estimate using 10^6 observations; (b)(c)(d) The predicted mean of p using 200, 800, and 2000 training samples, respectively.

3.6 Discussion and Conclusions

A probabilistic graphical model framework was developed to address the uncertainty propagation problem for flows in porous media. The framework could quantify the uncertainties propagating from the random input to the multi-output system response. The high dimensionality nature of the relationship between the inputs and responses was addressed by breaking the global problem into small local problems posed over coarse-elements. The whole framework was designed to be nonparametric (Gaussian mixture), so it was capable of capturing non-Gaussian features and thus it should have a wider applicabil-

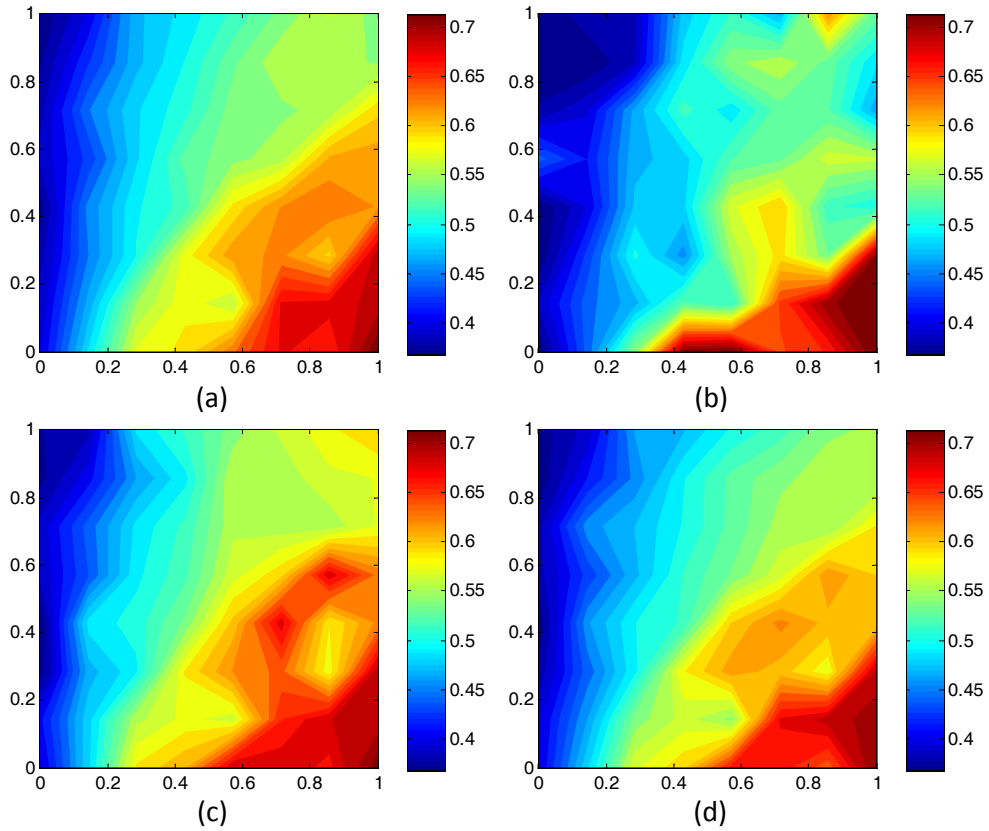

Figure 3.26: Non-stationary random field: Variance of u_x. (a) MC estimate using 10^6 observations; (b)(c)(d) The predicted variance of u_x using 200, 800, and 2000 training samples, respectively.

ity to other multiscale problems. The graphical model was shown that it can serve as a surrogate model for predicting the responses for any new observed permeability input.

Various examples were considered to study the accuracy and efficiency of the probabilistic graphical model framework and inference algorithms. It was shown that this framework is capable of predicting the correct output statistics with rather limited number of observations. In the provided examples, it was shown to capture well the first- and second-order statistics, and also provided reasonable predictions of the PDFs of the outputs. The framework can be used

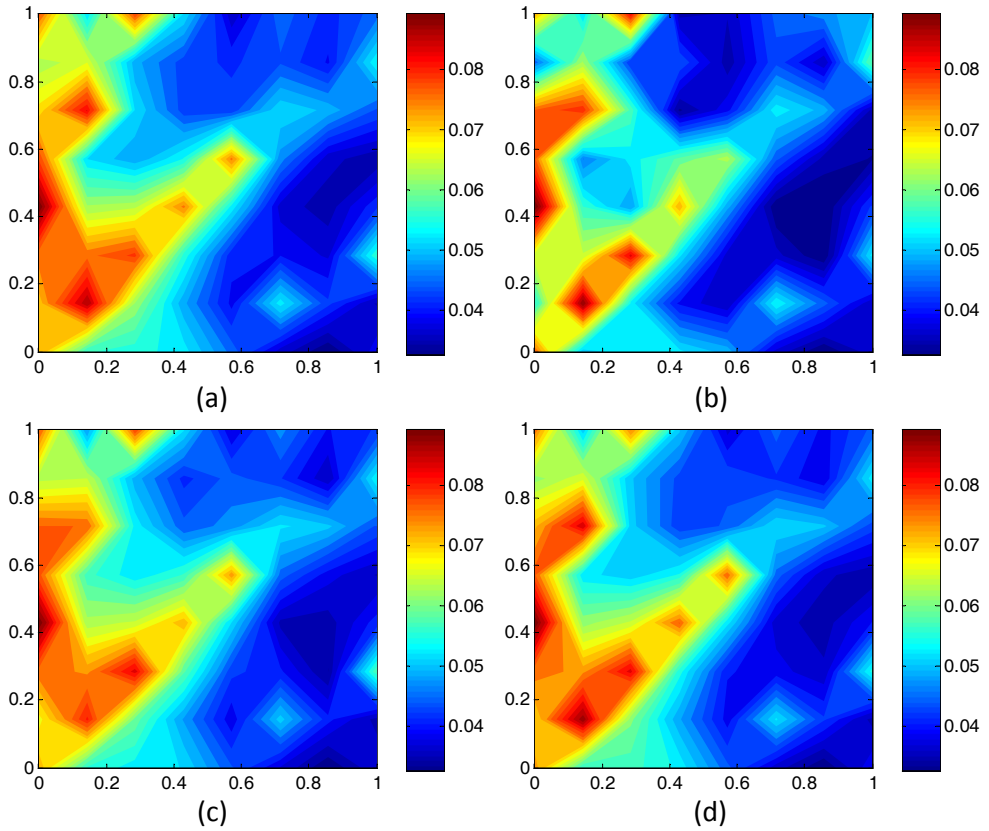

Figure 3.27: Non-stationary random field: Variance of u_y. (a) MC estimate using 10^6 observations; (b)(c)(d) The predicted variance of u_y using 200, 800, and 2000 training samples, respectively.

to address inverse problems (e.g. from limited output data predict unobservable permeability information). Such inverse problems and extending the applicability of the framework to other critical engineering applications are topics of current research interest.

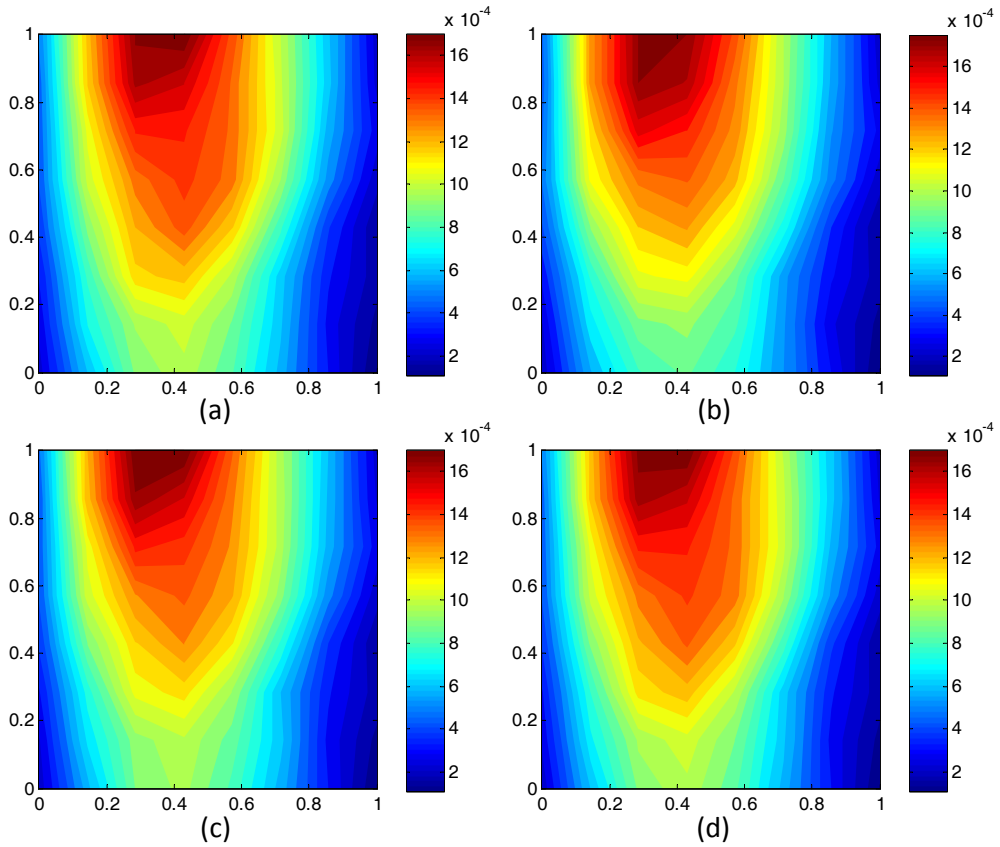

Figure 3.28: Non-stationary random field: Variance of p. (a) MC estimate using 10^6 observations; (b)(c)(d) The predicted variance of p using $200, 800$, and 2000 training samples, respectively.

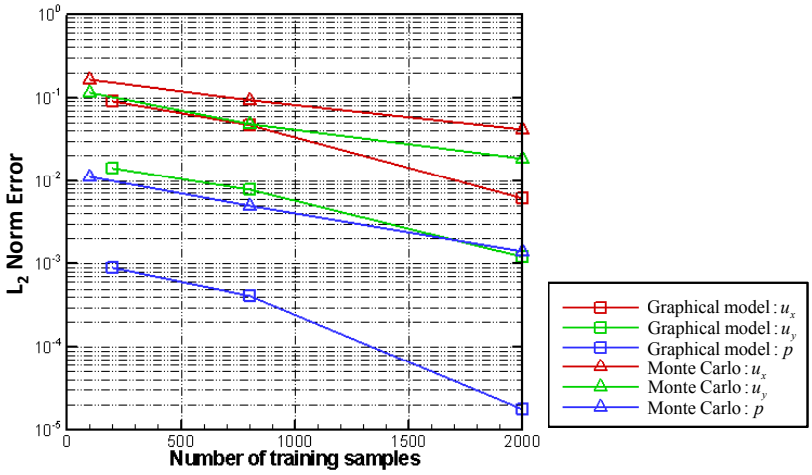

Figure 3.29: Non-stationary random field: The L_2 norm of the error as a function of the number of samples observed for graphical model framework.

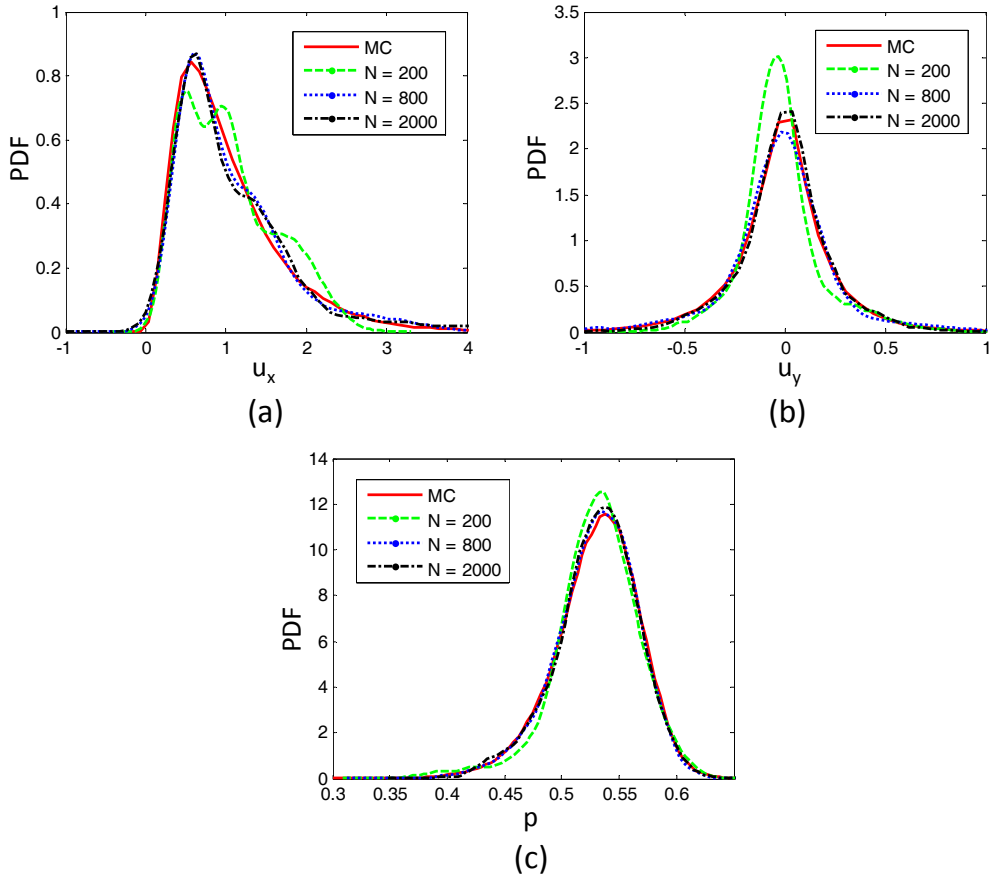

Figure 3.30: Non-stationary random field: Comparison of the predicted PDFs using different training data with the MC estimate at physical position $(0.429, 0.429)$ (a) u_x, (b) u_y, (c) p.

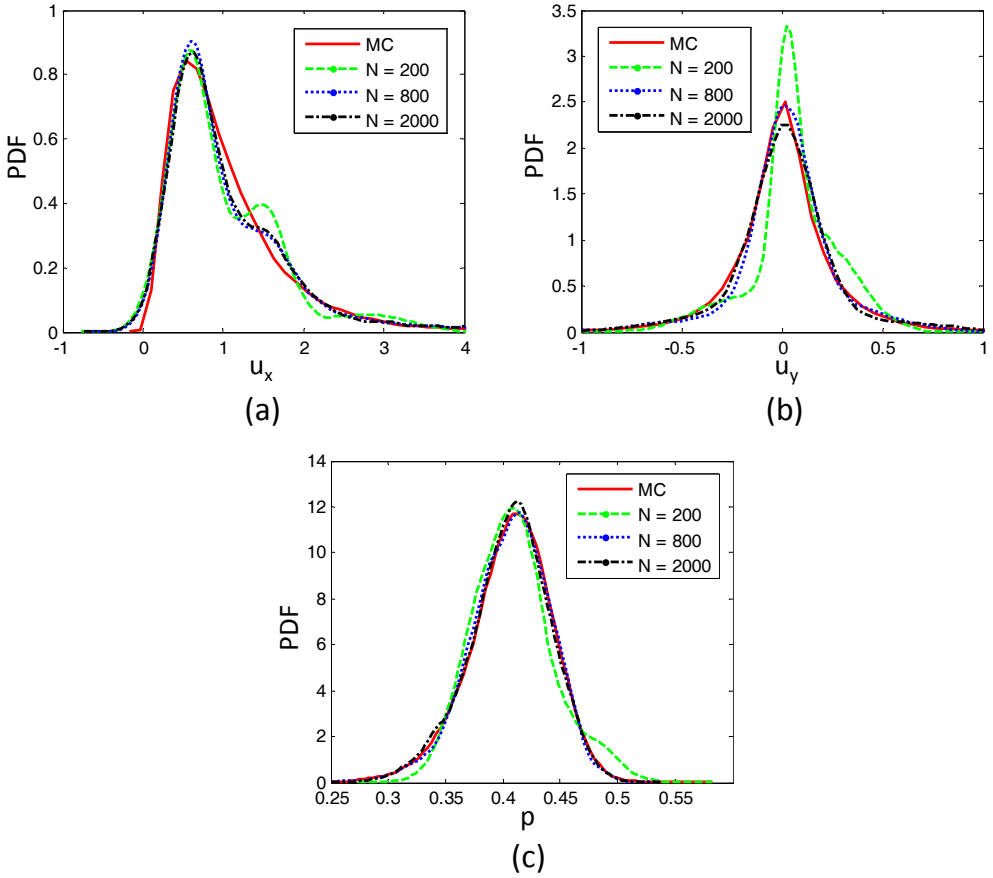

Figure 3.31: Non-stationary random field: Comparison of the predicted PDFs using different training data with the MC estimate at physical position $(0.571, 0.571)$ (a) u_x, (b) u_y, (c) p.

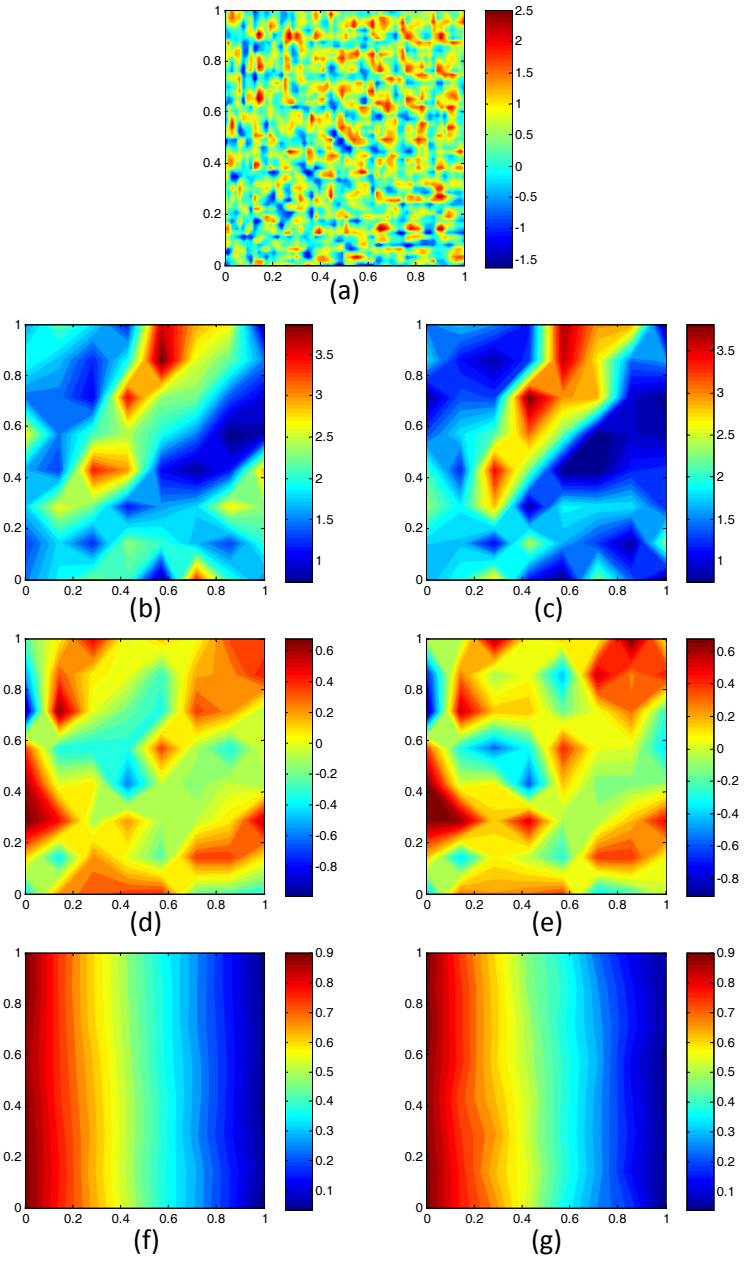

Figure 3.32: Non-stationary random field - Comparison of the predicted physical responses given a realization of stochastic input permeability with the true response: (a) The new observed input permeability field; (b)(d)(f) The true responses for the given permeability realization, from top to bottom, u_x, u_y and p, respectively; (c)(e)(g) The predicted means for u_x, u_y and p by graphical model using $N = 2000$ training data, respectively.

CHAPTER 4

UNCERTAINTY PROPAGATION USING INFINITE MIXTURE OF GAUSSIAN PROCESSES AND VARIATIONAL BAYESIAN INFERENCE

This chapter is organized as follows. In Section 4.1, we start by briefly presenting the core structure of our model and pay attention to each of the two main constituents: the MGP model (Section 4.1.1) and the DP prior (Section 4.1.2). In Section 4.1.4, we apply the VI methodology to our model and derive a fast approximation algorithm. Finally, in Section 4.1.5, we apply the constructed probabilistic surrogate surface to solve the UP problem. Numerical examples are presented in Section 4.2 demonstrating the high-accuracy and efficiency of the proposed framework. We conclude this work in Section 4.3.

4.1 Methodology

In general, the response of a physical model can be represented as a multi-output nonlinear function, $\mathbf{f} : \mathcal{X} \to \mathbb{R}^q$, where $\mathcal{X}$ denotes the input space and q is the number of output dimensions. The input space, $\mathcal{X}$, usually contains three distinct components: one for stochastics of the model, one for the spatial location, and one for time. The stochastics will be denoted by $\xi \in \mathcal{X}_\xi \subset \mathbb{R}^{d_\xi}$, and are treated as uncertain. The spatial location and the time variables will be denoted by $\mathbf{s} \in \mathcal{X}_s \subset \mathbb{R}^{d_s}$ with $d_s = 1, 2$ or 3 and $t \in \mathcal{X}_t = [0, T]$ with $T > 0$, respectively. That is, the input space, $\mathcal{X}$, is the Cartesian product:

$$\mathcal{X} = \mathcal{X}_\xi \times \mathcal{X}_s \times \mathcal{X}_t. \tag{4.1}$$

For notational convenience, we will refer to all the variables collectively by:

$$\mathbf{x} = (\boldsymbol{\xi}, \mathbf{s}, t). \tag{4.2}$$

The total dimensionality of the input space, $\mathcal{X}$, is $d = d_\xi + d_s + 1$. The output, $\mathbf{y} \in \mathbb{R}^q$, of the response function, $\mathbf{f}(\cdot)$, at input $\mathbf{x} = (\xi, \mathbf{s}, t)$,

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \mathbf{f}(\boldsymbol{\xi}, \mathbf{s}, t), \tag{4.3}$$

is thought as the value of the model at the spatial location $\mathbf{s}$ and time t when the generic uncertain inputs have the value ξ. The number q of the distinct outputs depends on the model. For example, when modeling a dynamical system, q would be equal to its dimension. For the 2D two-phase flow problem we will consider, the distinct outputs are the pressure, the x-velocity, the y-velocity, and the saturation of the two phases, i.e. $q = 4$. This framework is general enough to be applicable to most important problems.

In an UP problem, one assigns a probability density function (PDF) on the stochastic input, ξ, and wishes to quantify its effect on the output, $\mathbf{y}$, at some specific spatial locations and time instants. We will denote this PDF on ξ by $p(\xi)$. As noted in the introduction, this PDF usually represents our lack of knowledge about the true value of ξ, rather than intrinsic randomness. Now, the output of the model becomes a random field whose "randomness" corresponds to our uncertainty about the model's predictions. The goal of UP is to characterize the statistics of the output random field such as the *mean*,

$$\mathbf{m}_f(\mathbf{s}, t) = \int \mathbf{f}(\boldsymbol{\xi}, \mathbf{s}, t) p(\boldsymbol{\xi}) d\boldsymbol{\xi}, \tag{4.4}$$

the *covariance*,

$$\mathbf{C}_f(\mathbf{s}, t, \mathbf{s}', t') = \int \Big(\mathbf{f}(\boldsymbol{\xi}, \mathbf{s}, t) - \mathbf{m}_f(\mathbf{s}, t)\Big)\Big(\mathbf{f}(\boldsymbol{\xi}, \mathbf{s}', t') - \mathbf{m}_f(\mathbf{s}', t')\Big)^T p(\boldsymbol{\xi}) d\boldsymbol{\xi}, \tag{4.5}$$

the PDF of the output at a particular spatial location $\mathbf{s}_0$ and time instant t_0:

$$p(\mathbf{y}|s = \mathbf{s}_0, t = t_0) = \int \delta\left(\mathbf{f}(\xi, \mathbf{s}_0, t_0) - \mathbf{y}\right) p(\xi) d\xi, \tag{4.6}$$

where $\delta(\cdot)$ is Dirac's delta function, and so on.

Because of the computational complexity associated with evaluating the model, $\mathbf{f}(\cdot)$, the statistics outlined above cannot be computed directly. To remedy this situation, we resort to the usual two-step procedure of most UP methodologies. First, we construct a surrogate of $\mathbf{f}(\cdot)$, and, subsequently, we use it for the calculation of the statistics. The surrogate construction will be based on a set of observations $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$, where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times d}$ and $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^T \in \mathbb{R}^{N \times q}$ are the observed inputs and corresponding outputs, respectively. In particular, we consider a set of n_ξ observations of the stochastic inputs, $\Xi = (\xi_1, \ldots, \xi_{n_\xi})$, drawn from their PDF, $p(\xi)$. For each observed ξ, the model is evaluated on a preselected set of n_s spatial locations, $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_N)$, and n_t time instants, $\mathbf{t} = (t_1, \ldots, t_{n_t})$. That is, we have a total of $N = n_\xi n_s n_t$ observations in $\mathcal{D}$ available for the surrogate construction.

Our surrogate is going to be a Gaussian process [78], albeit a non-trivial one. Therefore, it can be thought of as a *Bayesian surrogate*, i.e. a probability measure on the space of plausible models that is compatible with the data $\mathcal{D}$. As shown in [11], the Bayesian nature of the surrogate can be exploited to derive error bars for any computed statistic. These error bars correspond to the epistemic uncertainty induced by the limited amount of data in $\mathcal{D}$. Similar ideas, can be used to solve inverse problems with very limited forward model evaluations [12]. This is one of the most important features of the Bayesian approach that makes it stand out from the traditional UP methodologies. This point will be further clarified in Section 4.1.5.

In an effort to capture non-stationary effects, such as localized features and discontinuities, we assume that the observed data are generated by an infinite number of different latent MGP's, $\left\{\mathbf{f}^{(m)}(\cdot)\right\}_{m=1}^{\infty}$. Each latent MGP, $\mathbf{f}^{(m)}(\cdot)$, explains a subset of the data. A Dirichlet process (DP) prior is used to generate the components of the infinite MGP mixture prior to observing any data. In practice, this mixture is truncated to M components. Our objective has three parts: find the optimal number of latent functions needed to explain a given set of data, cluster each data point on one of these models and train the m-th latent function, $\mathbf{f}^{(m)}(\cdot)$, using the data assigned to the m-th cluster. To achieve this, each observation is assigned a hidden variable $z_i \in \{1, 2, \ldots\}$ that classifies it. We collectively denote all these variables as $\mathbf{z} = (z_1, \ldots, z_N) \in \mathbb{N}^N$. Based on the assigned DP prior [80], each z_i follows a multinomial distribution. Finally, we are approximating the posterior of the model parameters by deriving a variational inference (VI) algorithm.

The outline of the remaining of this Section is as follows. In Section 4.1.1, we introduce the MGP model used for each one of the latent functions, $\mathbf{f}^{(m)}(\cdot)$. The Dirichlet process is discussed in Section 4.1.2 and the variational inference algorithm for the model is presented in Section 4.1.4. Finally, Section 4.1.5 discusses the integration of the model with uncertainty quantification tasks.

4.1.1 Multi-output Gaussian process regression

As mentioned earlier, each one of our latent functions, $\mathbf{f}^{(m)}(\cdot)$, is going to be a MGP [26, 11]. In this section, we briefly outline the specifics of this model. To keep the notation uncluttered, we will not be using the label m. It is implied on

every parameter used in this section.

The prior measure on latent functions Prior to seeing any data, the latent function, $\mathbf{f}(\cdot)$, is modeled as a q-dimensional Gaussian process:

$$\mathbf{f}(\cdot)|\mathbf{B}, \Sigma, \theta \sim \mathcal{N}_q\left(\mathbf{f}(\cdot)|\boldsymbol{\mu}\left(\cdot; \mathbf{B}\right), c\left(\cdot, \cdot; \theta\right) \Sigma\right), \tag{4.7}$$

conditional on the hyper-parameters $\mathbf{B}$, Σ, and θ. The symmetric, positive definite matrix $\Sigma \in \mathbb{R}^{q \times q}$, models the linear part of the correlations between the q distinct outputs. The *mean function*, $\mu\left(\cdot; \mathbf{B}\right)$, is given by:

$$\boldsymbol{\mu}\left(\mathbf{x}; \mathbf{B}\right) = \mathbf{B}^T \mathbf{h}(\mathbf{x}), \tag{4.8}$$

where $\mathbf{h}(\cdot) = (h_1(\cdot), \ldots, h_p(\cdot))$ are regression functions common to all outputs and $\mathbf{B} \in \mathbb{R}^{p \times q}$. For our numerical examples, we simply use linear regression functions:

$$\mathbf{h}(\cdot) = \left(1, \mathbf{x}^T\right).$$

The *covariance function*, $c\left(\cdot, \cdot; \theta\right)$, is taken to be:

$$c\left(\mathbf{x}, \mathbf{x}'; \theta\right) = \exp\left\{-\frac{1}{2} \sum_{l=1}^{d} \left(\frac{(x_l - x_l')^2}{(r_l)^2}\right)\right\} + \epsilon \delta\left(\mathbf{x} - \mathbf{x}'\right), \tag{4.9}$$

where $\theta = (r_1, \ldots, r_d, \epsilon) \in \mathbb{R}_+^{d+1}$. The parameters r_l can be interpreted as the length scales of the input dimension l, $l = 1, \ldots, d$. The parameter ϵ is known as the "nugget" and can be thought of as the variance of the noise of our model. Equation (4.7) defines a *probability measure* on the space of surrogates that corresponds to our prior beliefs.

The prior of the parameters To keep the notation concise, let us denote all the parameters of a latent function by ϕ:

$$\boldsymbol{\phi} = (\mathbf{B}, \Sigma, \theta). \tag{4.10}$$

The parameters ϕ take values in the space Ω_ϕ:

$$\Omega_\phi = \mathbb{R}^{p\times q} \times \mathbb{P}_q(\mathbb{R}) \times \mathbb{R}_+^{d+1}, \tag{4.11}$$

where $\mathbb{P}_q(\mathbb{R})$ is the space of q-dimensional positive-definite matrices with real coefficients. Following [26], we assign an *uninformative prior* on the pair $(\mathbf{B}, \Sigma)$ of the form:

$$\boldsymbol{p(\mathbf{B}, \Sigma) \propto |\Sigma|^{-\frac{q+1}{2}},} \tag{4.12}$$

and, following [11], an *exponential prior* on each of the components of θ:

$$\boldsymbol{p(\theta|\gamma) = \left[\prod_{i=1}^{d} \mathcal{E}\left(r_i|\gamma_r\right)\right] \mathcal{E}\left(\epsilon|\gamma_\epsilon\right),} \tag{4.13}$$

where $\gamma = (\gamma_r, \gamma_\epsilon)$ are the hyper-parameters of the exponential distribution $\mathcal{E}(\cdot|\gamma)$. The prior on ϕ is given by the product rule assuming that $(\mathbf{B}, \Sigma)$ and θ are a priori independent:

$$\boldsymbol{p(\phi|\gamma) = p(\mathbf{B}, \Sigma)p(\theta|\gamma).} \tag{4.14}$$

The likelihood of the data Now, assume that n observations have been classified to belong to the m-th cluster. Let $\mathbf{X}$ and $\mathbf{Y}$ be the corresponding inputs and outputs, respectively, of these n observations corresponding to the m-th cluster. They are collectively denoted by $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$. The *likelihood* of $\mathbf{Y}$ under our model (Eq. (4.7)) is given by the matrix-normal [27]:

$$\boldsymbol{\mathbf{Y}|\mathbf{X}, \phi \sim \mathcal{N}_{n\times q}\left(\mathbf{Y}|\mathbf{HB}, \mathbf{A}, \Sigma\right),} \tag{4.15}$$

where $\mathbf{H} \in \mathbb{R}^{n\times p}$ is the *design matrix*,

$$H_{ij} = h_j(\mathbf{x}_i), \tag{4.16}$$

and $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the *covariance matrix*,

$$A_{ij} = c\left(\mathbf{x}_i, \mathbf{x}_j; \theta\right). \tag{4.17}$$

The posterior of the parameters The *posterior* of the parameters (conditioned on γ) is given by:

$$p\left(\phi|\mathcal{D},\gamma\right)=\frac{p\left(\mathbf{Y}|\mathbf{X},\phi\right)p\left(\phi|\gamma\right)}{p\left(\mathcal{D}|\gamma\right)}, \tag{4.18}$$

where $p\left(\mathbf{Y}|\mathbf{X},\phi\right)$ is the likelihood given in Eq. (4.15) and $p\left(\mathcal{D}|\gamma\right)$ is the *evidence* given by:

$$\boldsymbol{p\left(\mathcal{D}|\gamma\right)=\int p\left(\mathbf{Y}|\mathbf{X},\phi\right)p\left(\phi|\gamma\right)d\phi.} \tag{4.19}$$

The derivation of the posterior $p\left(\mathbf{B},\Sigma,\theta|\mathcal{D}\right)$ is given in Appendix C.1.

The posterior probability measure on latent functions With the prior probability measure of Eq. (4.7) and conditioning on the observed data $\mathcal{D}$, we derive the *posterior probability measure* on the space of surrogates:

$$\mathbf{f}(\cdot)|\boldsymbol{\phi}\sim\mathcal{N}_q\left(\mathbf{f}(\cdot)|\boldsymbol{\mu}^*\left(\cdot;\mathbf{B}\right),\boldsymbol{c}^*\left(\cdot,\cdot;\boldsymbol{\theta}\right)\boldsymbol{\Sigma}\right), \tag{4.20}$$

where the *posterior mean function* is given by:

$$\boldsymbol{\mu}^*\left(\mathbf{x};\mathbf{B}\right)=\mathbf{B}^T\mathbf{h}\left(\mathbf{x}\right)+\left(\mathbf{Y}-\mathbf{HB}\right)^T\mathbf{A}^{-1}\mathbf{a}\left(\mathbf{x}\right), \tag{4.21}$$

and the *posterior covariance function* by:

$$c^*\left(\mathbf{x},\mathbf{x}';\theta\right)=c\left(\mathbf{x},\mathbf{x}';\theta\right)-\mathbf{a}\left(\mathbf{x}\right)^T\mathbf{A}^{-1}\mathbf{a}\left(\mathbf{x}'\right), \tag{4.22}$$

with $\mathbf{a}(\cdot)=\left(c\left(\cdot,\mathbf{x}_1;\theta\right),\ldots,c\left(\cdot,\mathbf{x}_n;\theta\right)\right)\in\mathbb{R}^n$. Eq. (4.20) can be used to draw samples of candidate surrogates that are compatible with the data $\mathcal{D}$ as well as our prior beliefs if all the hyper-parameters are known. Alternatively, its mean function could be used as surrogate surface in the classical way.

Integrating out B and Σ If $n\geq p+q$ (p is the number of regression functions and q the number of output dimensions), then the choice of prior we made

in Eq. (4.12) for the pair $(\mathbf{B}, \Sigma)$ allows us to integrate it out from the posterior (see [26] for the details). In particular, multiplying Eq. (4.20) with Eq. (4.18) and integrating out $\mathbf{B}$ and Σ, yields the posterior measure conditioned only on θ which is a q-dimensional student-$\mathcal{T}$ process:

$$\mathbf{f}(\cdot)|\theta, \mathcal{D} \sim \mathcal{T}_q\left(\mathbf{f}(\cdot)|\mu^{**}\left(\cdot; \widehat{\mathbf{B}}\right), c^{**}(\cdot, \cdot; \theta)\, \widehat{\Sigma}; n - p\right), \tag{4.23}$$

with the mean and covariance functions:

$$\begin{aligned} \mu^{**}(\mathbf{x}) &= \mu^{*}\left(\mathbf{x}; \widehat{\mathbf{B}}\right), & (4.24)\\ c^{**}(\mathbf{x}, \mathbf{x}'; \theta) &= c^{*}(\mathbf{x}, \mathbf{x}'; \theta) + \\ &\quad \left(\mathbf{h}(\mathbf{x}) - \mathbf{H}^T\mathbf{A}^{-1}\mathbf{a}(\mathbf{x})\right)^T \left(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H}\right)^{-1} \left(\mathbf{h}(\mathbf{x}') - \mathbf{H}^T\mathbf{A}^{-1}\mathbf{a}(\mathbf{x}')\right), & (4.25) \end{aligned}$$

where

$$\begin{aligned} \widehat{\mathbf{B}} &= \left(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{A}^{-1}\mathbf{Y}, & (4.26)\\ \widehat{\Sigma} &= \frac{1}{n-p}\left(\mathbf{Y} - \mathbf{H}\widehat{\mathbf{B}}\right)^T \mathbf{A}^{-1}\left(\mathbf{Y} - \mathbf{H}\widehat{\mathbf{B}}\right). & (4.27) \end{aligned}$$

This approach will be used for making predictions with each one of our latent models.

The posterior of $(\mathbf{B}, \Sigma)$ conditioned on θ The posterior of $(\mathbf{B}, \Sigma)$ has an analytic form:

$$p(\mathbf{B}, \Sigma|\mathcal{D}, \theta) = \mathcal{N}_{p\times q}\left(\mathbf{B}|\widehat{\mathbf{B}}, (\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})^{-1}, \Sigma\right) \mathcal{W}_q^{-1}\left(\Sigma|(n-p)\widehat{\Sigma}, n-p\right), \tag{4.28}$$

where $\mathcal{W}_q^{-1}(\cdot|\mathbf{W}, \nu)$ is the Inverse-Wishart distribution [57] with scale matrix $\mathbf{W} \in \mathbb{P}_q(\mathbb{R})$ and ν degrees of freedom. This result would be useful in the variational formulation because it specifies the form that a candidate posterior for the pair $(\mathbf{B}, \Sigma)$ should take, as dicussed in Section 4.1.4. The derivation of Eq. (4.28) and also of $p(\mathbf{B}|\mathcal{D}, \Sigma, \theta)$, $p(\Sigma|\mathcal{D}, \theta)$, and $p(\theta|\mathcal{D})$, are given in Appendix C.1.

Remark 1. Alternative choices to the prior in Eq. (4.12) can be selected. For example, a conjugate prior for $\mathbf{B}$ and Σ can be taken as $p(\mathbf{B}|\Sigma) = \mathcal{N}_{p\times q}(\mathbf{B}; \mathbf{0}, \mathbb{I}_p, \Sigma)$ and $p(\Sigma) = \mathcal{W}_q^{-1}(\Sigma; \mathbb{I}_q, q)$, where $\mathbb{I}_p$ is the identity matrix $q \times q$ matrix. With this prior choice, the posterior of $(\mathbf{B}, \Sigma)$ conditional on θ can be shown to be:

$$\boldsymbol{p(\mathbf{B}, \Sigma|\mathcal{D}, \theta) = \mathcal{N}_{p\times q}\left(\mathbf{B}|\widehat{\mathbf{B}}, (\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H} + \mathbb{I}_p)^{-1}, \Sigma\right)\mathcal{W}_q^{-1}\left(\Sigma|(n+q)\widehat{\Sigma}, n+q\right)}, \quad (4.29)$$

where

$$\widehat{\mathbf{B}} = \left(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H} + \mathbb{I}_p\right)^{-1}\mathbf{H}^T\mathbf{A}^{-1}\mathbf{Y}, \quad (4.30)$$

$$\widehat{\Sigma} = \frac{1}{n+q}\left[\left(\mathbf{Y} - \mathbf{H}\widehat{\mathbf{B}}\right)^T\mathbf{A}^{-1}\left(\mathbf{Y} - \mathbf{H}\widehat{\mathbf{B}}\right) + \widehat{\mathbf{B}}^T\widehat{\mathbf{B}} + \mathbb{I}_q\right]. \quad (4.31)$$

In comparison to Eq. (4.28) that was derived using the prior in Eq. (4.12), note that the posterior above differs by the presence of some additional bias terms in $\widehat{\mathbf{B}}$ and $\widehat{\Sigma}$ (compare Eqs. (4.30) and (4.31) to Eqs. (4.26) and (4.27), respectively) and a different degree of freedom for the Inverse-Wishart distribution for Σ. The additional bias terms ensure non-singularity in the affiliated matrix calculations, e.g., $\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H}$ is a singular matrix if $n < p$, but $(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H} + \mathbb{I}_p)$ can always be inverted regardless of the number of observations for each component $\mathbf{f}^{(m)}(\cdot)$. However, the existence of the additional bias terms affects the accuracy of local models with few observations. Therefore, we choose not to use such kind of priors in this paper.

4.1.2 The Dirichlet process

A DP [32] defines a probability measure on a space of probability measures. In particular, let Ω be a set and $\mathcal{F}$ be a σ-algebra on Ω. The space of probability

measures on $(\Omega, \mathcal{F})$ is denoted by:

$$\mathcal{P}(\Omega, \mathcal{F}) = \left\{ G : \mathcal{F} \to \mathbb{R}_+ \text{ is a probability measure } \right\}. \tag{4.32}$$

A DP defines a probability measure on $\mathcal{P}(\Omega, \mathcal{F})$. That is, a sample $G(\cdot)$ from a DP is in $\mathcal{P}(\Omega, \mathcal{F})$.

Let $G_0(\cdot) \in \mathcal{P}(\Omega, \mathcal{F})$ and $\alpha_0 > 0$ some constant. Then $G(\cdot)$ is a sample from the DP induced by $G_0(\cdot)$ and α_0, i.e.

$$G(\cdot) \sim \mathcal{D}P\left(G_0(\cdot), \alpha_0\right), \tag{4.33}$$

if and only if for any partition $\{A_1, \ldots, A_k\} \subset \mathcal{F}$ of Ω we have:

$$(G(A_1), \ldots, G(A_k)) \sim \mathcal{D}ir_k\left(G(A_1), \ldots, G(A_k) | \alpha_0 G_0(A_1), \ldots, \alpha_0 G_0(A_k)\right), \tag{4.34}$$

where $\mathcal{D}ir_k(\cdot, \ldots, \cdot | a_1, \ldots, a_k)$ denotes the Dirichlet distribution with parameters $a_1, \ldots, a_k$.

The "stick-breaking" construction The "stick-breaking" construction of [90] allows us to define the DP in a generative way by the use of some intermediate random variables. In particular, let $G_0(\cdot)$ and α_0 be as before and define the random variables $\nu = (\nu_1, \nu_2, \ldots) \in [0, 1]^\infty$ by:

$$\boldsymbol{\nu | \alpha_0 \sim \prod_{m=1}^{\infty} \mathcal{B}eta\left(\nu_m | 1, \alpha_0\right),} \tag{4.35}$$

the sequence of numbers:

$$\boldsymbol{\pi_m(\nu) = \nu_m \prod_{i=1}^{m-1} (1 - \nu_i),} \tag{4.36}$$

and the random variables $\omega = (\omega_1, \omega_2, \ldots)$ by:

$$\boldsymbol{\omega | G_0(\cdot) \sim \prod_{m=1}^{\infty} G_0(\omega_m).} \tag{4.37}$$

Then, that the random probability measure on $(\Omega, \mathcal{F})$ defined by:

$$\boldsymbol{G(\cdot; \nu, \omega) = \sum_{m=1}^{\infty} \pi_m(\nu)\delta_{\omega_m}(\cdot),} \tag{4.38}$$

where $\delta_{\omega_m}(\cdot)$ is Dirac's delta function centered at ω_m, is a sample from the DP defined in Eq. (4.33).

4.1.3 An infinite mixture of MGP's using the Dirichlet Process

According to the discussion of Sec. 4.1.1, each latent MGP model is characterized uniquely by choosing its parameters $\phi \in \Omega_\phi$. The DP concept is used to define a prior probability measure on the space of probability measures of the model space Ω_ϕ. The role of this DP is to generate the components of the MGP mixture prior to observing any data.

Let α_0 be as before, $\mathcal{F}_\phi$ be a σ-algebra on Ω_ϕ, and $P_{\phi,0}(\cdot|\gamma)$ be the probability measure on $(\Omega_\phi, \mathcal{F}_\phi)$ induced by the prior $p(\phi|\gamma)$ of Eq. (4.14), i.e. for each $A \in \mathcal{F}_\phi$ we have:

$$\boldsymbol{P_{\phi,0}(A|\gamma) = \int_A p(\phi|\gamma)d\phi.} \tag{4.39}$$

The Dirichlet process $\mathcal{D}P(\cdot|P_{\phi,0}(\cdot;\gamma), \alpha_0)$ is used to define a prior probability measure on $\mathcal{P}(\Omega_\phi, \mathcal{F}_\phi)$. A sample $P_\phi(\cdot)$ from $\mathcal{D}P(\cdot|P_{\phi,0}(\cdot;\gamma), \alpha_0)$ is then used to generate the mixture components. Let ν and $\pi_m(\nu), m = 1, 2, \ldots$ be as in Eq. (4.35) and Eq. (4.36), respectively, and, as in Eq. (4.37), let

$$\boldsymbol{\Phi = (\phi_1, \phi_2, \ldots)} \tag{4.40}$$

be given by:

$$\boldsymbol{\Phi|\gamma \sim \prod_{m=1}^{\infty} P_{\phi,0}(\phi_m|\gamma) = \prod_{m=1}^{\infty} p(\phi_m|\gamma).} \tag{4.41}$$

Notice that, because the measure $P_{\phi,0}$ is absolutely continuous (see Eq. (4.39)), it is guaranteed that all the ϕ_m's generated in Eq. (4.41) are distinct. Making use of the "breaking-stick" construction, we see that a sample from $\mathcal{DP}(\cdot|P_{\phi,0}(\cdot;\gamma),\alpha_0)$ can now be represented (see Eq. (4.38)) as:

$$\boldsymbol{P_\phi(\cdot;\nu,\Phi)=\sum_{m=1}^{\infty}\pi_m(\nu)\delta_{\phi_m}(\cdot).} \tag{4.42}$$

Each observation $(\mathbf{x},\mathbf{y})$ is generated by sampling one of the models $m=1,\ldots,\infty$. Each model m is uniquely characterized by the parameters ϕ_m. These parameters need to be sampled from Eq. (4.42) (the sample from the GP). Because Eq. (4.42) is degenerate, we will necessarily obtain one of the ϕ_m's we draw from the prior in Eq. (4.41). Thus to assign a model to $(\mathbf{x},\mathbf{y})$, one needs to simply pick a number $m=1,\ldots,\infty$ by sampling the multinomial distribution with probabilities $\pi_m(\nu)$. Rather than doing this directly, is is more convenient to introduce a latent (indicator) variable z whose sole role is to pick a number from $m=1,\ldots,\infty$ with the right probabilities (Eq. (4.42)). Then, if we draw a z, we can say that ϕ_z (the ϕ_m's are coming from Eq. (4.41)) is the model associated with $(\mathbf{x},\mathbf{y})$.

For N observations, we thus introduce *indicator variables* $z_i\in\mathbb{N}$, for each $i=1,\ldots,N$. Let $\mathbf{z}\in\mathbb{N}^N$ be the vector of indicator variables, called the *indicator vector*:

$$\mathbf{z}=(z_1,\ldots,z_N). \tag{4.43}$$

The prior assigned to $\mathbf{z}$ (following Eq. (4.42)) is taken as:

$$\begin{aligned} p(\mathbf{z}|\nu) &= \prod_{n=1}^{N}\sum_{m=1}^{\infty}\pi_m(\nu)^{\mathbf{1}[z_n=m]} \\ &= \prod_{i=1}^{N}\mathcal{M}ulti\,(z_n|\pi_1(\nu),\pi_2(\nu),\ldots)=\prod_{n=1}^{N}\pi_{z_n}(\nu), \end{aligned} \tag{4.44}$$

where $\mathcal{M}ulti(\cdot|\pi_1,\pi_2,\ldots)$ stands for the multinomial probability distribution with probabilities $\pi_1,\pi_2,\ldots$. Observe that there are not necessarily N distinct latent models as $\phi_n^*=\phi_{z_n}$ are distributed according to $P_\phi(\cdot;\nu,\mathbf{\Phi})(\cdot)$ of Eq. (4.42).

Following the developments of the gating network in [77], we consider the indicator vector to be dependent on the input:

$$p(\mathbf{z}|\mathbf{X}, \mathbf{m}, \mathbf{R}, \nu) = \prod_{n=1}^{N} \prod_{m=1}^{\infty} \left(\frac{p(\mathbf{x}_n|\mathbf{m}_m, \mathbf{R}_m)\pi_m(\nu)}{\sum_{j=1}^{\infty} p(\mathbf{x}_n|\mathbf{m}_j, \mathbf{R}_j)\pi_j(\nu)} \right)^{\mathbf{1}[z_n=m]}, \tag{4.45}$$

where $\pi_m(\nu)$ is the prior on z defined above in Eq. (4.36) and $p(\mathbf{x}_n|\mathbf{m}_m, \mathbf{R}_m)$ is defined by a Gaussian function as [95]:

$$p(\mathbf{x}_n|z_n = m, \mathbf{m}_m, \mathbf{R}_m) = \mathcal{N}_d(\mathbf{x}_n|\mathbf{m}_m, \mathbf{R}_m^{-1}), \tag{4.46}$$

with $\mathbf{m}_m$, $\mathbf{R}_m$ its mean and precision matrix, respectively. We denote $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots\}$ and $\mathbf{R} = \{\mathbf{R}_1, \mathbf{R}_2, \ldots\}$ the mean and precision for all mixture components. Conjugate priors to $\mathbf{m}_m$ and $\mathbf{R}_m$ are assigned as

$$\mathbf{m}_m = \mathcal{N}_d\left(\mathbf{u}_0, \mathbf{R}_0^{-1}\right), \tag{4.47}$$

$$\mathbf{R}_m = \mathcal{W}_d\left(\mathbf{W}_0, \upsilon_0\right). \tag{4.48}$$

Remark 2**.** In addressing the input clustering problem, we consider a full precision matrix $\mathbf{R}_m$ that governs the size and shape of each local model m. With such a choice, both the inner- and inter-correlations between the different input dimensions $(\xi, \mathbf{s}, t)$ are considered. A simplification of the clustering model above can be introduced by considering a diagonal precision matrix $\mathbf{R}_m$ with each diagonal element related to the correlation length parameters of the MGP model (Eq. (4.9)).

To finalize the description of the model, we define an operator that looks at $\mathbf{z}$ and clusters the data $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$ according to the latent function that explains them. Towards this goal, define the *selection operator* for any matrix $\mathbf{W} \in \mathbf{R}^{n\times\ell}$ by:

$$\mathcal{S}_m(\mathbf{W}; \mathbf{z}) := \{\mathbf{w}_n : \text{ if } z_n = m\}, \tag{4.49}$$

where $\mathbf{w}_n$ is the n-th row of $\mathbf{W}$. $\mathcal{S}_m(\mathbf{W}|\mathbf{z})$ is to be thought of as a matrix with $|z_n = m|$ rows and ℓ columns ($|B|$ is the number of elements of the set B). Using this definition, the inputs and outputs associated with the latent function m are $\mathcal{S}_m(\mathbf{X};\mathbf{z})$ and $\mathcal{S}_m(\mathbf{Y};\mathbf{z})$, respectively.

The *likelihood* of the full model can be described as:

$$\boldsymbol{p}(\mathbf{Y},\mathbf{z}|\mathbf{X},\boldsymbol{\Phi}) = \boldsymbol{p}(\mathbf{Y}|\mathbf{X},\mathbf{z},\boldsymbol{\Phi})\,\boldsymbol{p}(\mathbf{z}|\mathbf{X},\mathbf{m},\mathbf{R},\boldsymbol{\nu}), \tag{4.50}$$

where $p(\mathbf{Y}|\mathbf{X},\mathbf{z},\Phi)$ is given as:

$$p(\mathbf{Y}|\mathbf{X},\mathbf{z},\Phi) = \prod_{\{m:\,|\{z_n=m\}|>0\}} p\left(\mathcal{S}_m(\mathbf{Y};\mathbf{z})\,|\mathcal{S}_m(\mathbf{X};\mathbf{z}),\phi_m\right), \tag{4.51}$$

where the product is over latent functions that are associated with at least one observation. The likelihood term pertaining to the m-th latent function, $p\left(\mathcal{S}_m(\mathbf{Y};\mathbf{z})\,|\mathcal{S}_m(\mathbf{X};\mathbf{z}),\phi_m\right)$, is given by Eq. (4.15) with $\mathcal{S}_m(\mathbf{X};\mathbf{z}), \mathcal{S}_m(\mathbf{Y};\mathbf{z})$, and ϕ_m instead of $\mathbf{X}$, $\mathbf{Y}$, and ϕ, respectively.

The *prior* of the full model is:

$$\boldsymbol{p}(\boldsymbol{\Phi},\boldsymbol{\nu},\mathbf{m},\mathbf{R}|\mathcal{I}) = \boldsymbol{p}(\boldsymbol{\Phi}|\boldsymbol{\gamma})\boldsymbol{p}(\boldsymbol{\nu}|\boldsymbol{\alpha_0})\boldsymbol{p}(\mathbf{m}|\mathbf{u_0},\mathbf{R_0})\boldsymbol{p}(\mathbf{R}|\mathbf{W_0},\boldsymbol{\upsilon_0}), \tag{4.52}$$

where we define the hyper-parameters as $\mathcal{I} = (\gamma, \alpha_0, \mathbf{u}_0, \mathbf{R}_0, \mathbf{W}_0, \upsilon_0)$. $p(\Phi|\gamma)$, and $p(\nu|\alpha_0)$ are given by Eqs. (4.41) and (4.35), respectively, and $p(\mathbf{m}|\mathbf{u}_0,\mathbf{R}_0)$ and $p(\mathbf{R}|\mathbf{W}_0,\upsilon_0)$ are given by Eqs. (4.47) and (4.48), respectively.

The *posterior* of the full model is:

$$\boldsymbol{p}(\boldsymbol{\Phi},\mathbf{z},\boldsymbol{\nu},\mathbf{m},\mathbf{R}|\mathcal{D},\mathcal{I}) = \frac{\boldsymbol{p}(\mathbf{Y}|\mathbf{X},\mathbf{z},\boldsymbol{\Phi})\,\boldsymbol{p}(\mathbf{z}|\mathbf{X},\mathbf{m},\mathbf{R},\boldsymbol{\nu})\boldsymbol{p}(\boldsymbol{\Phi},\boldsymbol{\nu},\mathbf{m},\mathbf{R}|\mathcal{I})}{\boldsymbol{p}(\mathcal{D}|\mathcal{I})}, \tag{4.53}$$

where $p(\mathbf{Y}|\mathbf{X},\mathbf{z},\Phi)$, $p(\mathbf{z}|\mathbf{X},\mathbf{m},\mathbf{R},\nu)$ and $p(\Phi,\nu,\mathbf{m},\mathbf{R}|\mathcal{I})$ are given in Eqs. (4.51), (4.45) and (4.52), respectively, and the evidence is:

$$\boldsymbol{p}(\mathcal{D}|\mathcal{I}) = \sum_{\mathbf{z}} \int \boldsymbol{p}(\mathbf{Y}|\mathbf{X},\mathbf{z},\boldsymbol{\Phi})\,\boldsymbol{p}(\mathbf{z}|\mathbf{X},\mathbf{m},\mathbf{R},\boldsymbol{\nu})\boldsymbol{p}(\boldsymbol{\Phi},\boldsymbol{\nu},\mathbf{m},\mathbf{R}|\mathcal{I})\boldsymbol{d\Phi}\boldsymbol{d\nu}\boldsymbol{d}\mathbf{m}\boldsymbol{d}\mathbf{R}. \tag{4.54}$$

A probabilistic graphical model representation of this model is given in Fig. 4.1.

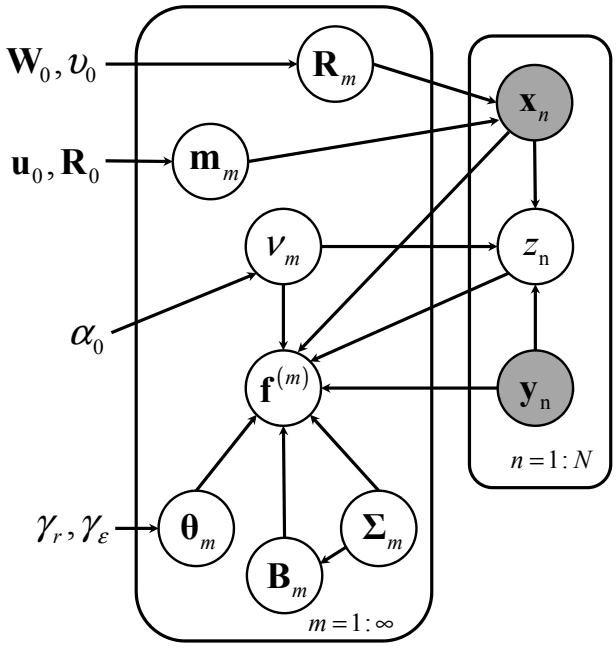

Figure 4.1: Graphical model representation of the introduced framework. The input variable $\mathbf{X}$ is determined by ξ, $\mathbf{S}$ and $\mathbf{t}$. $\mathbf{x}_n$ and $\mathbf{y}_n, n = 1, \ldots, N$ are the observations. $\mathbf{f}^{(m)}$ denotes the m-th MGP model, ν_m, $\theta^{(m)}$, $\mathbf{B}^{(m)}$, and $\Sigma^{(m)}$ are the affiliated parameters to $\mathbf{f}^{(m)}$, and α_0, γ_r and γ_ϵ are the hyperparameters. $\mathbf{m}_m$ and $\mathbf{R}_m$ are parameters used to determine the clustering of observations, and $\mathbf{u}_0$, $\mathbf{R}_0$, $\mathbf{W}_0$, and υ_0 are the corresponding hyperparameters. z_n is the hidden variable that classifies each observation. Based on the classification, $\mathbf{f}^{(m)}$ is constructed using only the m-th data subset.

4.1.4 Variational inference

VI aims at obtaining an approximation of the posterior (see Eq. (4.53)) by transforming the inference problem to an optimization problem. In particular, it involves minimizing the "distance" between the true posterior of Eq. (4.53) from a family of candidate probability distributions [14].

To simplify the notation, let us denote both the latent variables $\mathbf{z}$ and ν, the input parameters $\mathbf{m}, \mathbf{R}$, and the parameters of the model Φ as follows:

$$\boldsymbol{\Psi} = \{\mathbf{z}, \nu, \mathbf{m}, \mathbf{R}, \boldsymbol{\Phi}\}\,. \tag{4.55}$$

We seek an approximation $q(\Psi)$ to the true posterior of the model $p(\Psi|\mathcal{D}, \mathcal{I})$ (see Eq. (4.53)) by minimizing the Kullback–Leibler (KL) divergence:

$$\mathbf{KL}[q \parallel p] = \int q(\boldsymbol{\Psi}) \ln \frac{q(\boldsymbol{\Psi})}{p(\boldsymbol{\Psi}|\mathcal{D}, \mathcal{I})} d\boldsymbol{\Psi}. \tag{4.56}$$

The KL divergence can be thought of as a "distance" between two probability distributions. It is always greater than or equal to zero and is exactly zero when the two distributions coincide. Dealing with the posterior directly is impossible because the evidence $p(\mathcal{D}|\mathcal{I})$ of Eq. (4.54) is not known. However, notice that (see Appendix C.2):

$$\ln p(\mathcal{D}|\mathcal{I}) = \mathrm{KL}[q \parallel p] + \mathcal{L}[q], \tag{4.57}$$

where $\mathcal{L}[q]$ is given by:

$$\mathcal{L}[q] = \int q(\boldsymbol{\Psi}) \ln \frac{p(\mathcal{D}, \boldsymbol{\Psi}|\mathcal{I})}{q(\boldsymbol{\Psi})} d\boldsymbol{\Psi}. \tag{4.58}$$

$\mathcal{L}[q]$ is a lower bound to the logarithm of the evidence and depends only on the joint distribution of Ψ and $\mathcal{D}$, $p(\mathcal{D}, \Psi|\mathcal{I})$. Therefore, minimizing Eq. (4.56) is equivalent to maximizing $\mathcal{L}[q]$. Solving this maximization problem is the goal of the remaining of this section.

The joint distribution $p(\mathcal{D}, \Psi|\mathcal{I})$ is represented as:

$$\begin{aligned}
& p(\mathcal{D}, \Psi|\mathcal{I}) = p\left(\mathbf{Y}|\mathbf{X}, \mathbf{z}, \Phi\right) p(\mathbf{z}|\mathbf{X}, \mathbf{m}, \mathbf{R}, \nu) p(\Phi, \nu, \mathbf{m}, \mathbf{R}|\mathcal{I}) \\
= & \prod_{m=1}^{\infty} \left\{ p\left(\mathcal{S}_m\left(\mathbf{Y}; \mathbf{z}\right)|\mathcal{S}_m\left(\mathbf{X}; \mathbf{z}\right), \phi_m\right) \prod_{n=1}^{N} \left(\frac{p(\mathbf{x}_n|\mathbf{m}_m, \mathbf{R}_m)\pi_m(\nu)}{\sum_{j=1}^{\infty} p(\mathbf{x}_n|\mathbf{m}_j, \mathbf{R}_j)\pi_j(\nu)} \right)^{\mathbf{1}[z_n=m]} \right\} p(\Phi, \nu, \mathbf{m}, \mathbf{R}|\mathcal{I}) \\
= & \prod_{m=1}^{\infty} \left\{ p\left(\mathcal{S}_m\left(\mathbf{Y}; \mathbf{z}\right)|\mathcal{S}_m\left(\mathbf{X}; \mathbf{z}\right), \phi_m\right) \prod_{n=1}^{N} \left(\frac{p(\mathbf{x}_n|\mathbf{m}_m, \mathbf{R}_m)\pi_m(\nu)}{C_n(\mathbf{m}, \mathbf{R}, \nu)} \right)^{\mathbf{1}[z_n=m]} \right\} p(\Phi, \nu, \mathbf{m}, \mathbf{R}|\mathcal{I}),
\end{aligned} \tag{4.59}$$

where we define $C_n(\mathbf{m}, \mathbf{R}, \nu) = \sum_{j=1}^{\infty} p(\mathbf{x}_n|\mathbf{m}_j, \mathbf{R}_j)\pi_j(\nu)$. The normalization constant depends on the parameters $\mathbf{m}, \mathbf{R}$, and ν and contributes to the variational updates of the posterior distributions. However to simplify the variational inference algorithm, we follow the spirit of the EM algorithm [14] by using $\widehat{C}_n(\widehat{\mathbf{m}}, \widehat{\mathbf{R}}, \widehat{\nu})$ to approximate $C_n(\mathbf{m}, \mathbf{R}, \nu)$, where $\widehat{\mathbf{m}}, \widehat{\mathbf{R}}$, and $\widehat{\nu}$ are the means of the corresponding posterior distributions. In practice, these values are obtained from the approximated posterior distributions in the previous iteration step of the variational inference algorithm.

Remark 3**.** Another possible way of approximating this normalization term is by finding an upper bound of it [17]. In the context of mixture of expert models, these bounds have been shown to allow straightforward implementation of variational inference for computing approximations to the posterior of the model parameters. However, application of such bounds to our infinite mixture of MGP models using Eq. (4.45) (note the presence of the $\pi_m(\nu)$ term inside the summation in the normalization factor) leads to non-standard distributions that can only be approximated by sampling or non-parametric (Gaussian mixture) forms. In addition to the unknown parameters $\mathbf{m}, \mathbf{R}$, these bounds depend in a complicated manner on ν thus not allowing closed form approximations or a computationally efficient implementation.

Following [95, 15], we are looking for solutions that factorize as follows:

$$q(\mathbf{\Psi}) = \prod_{n=1}^{N} q(z_n) \prod_{m=1}^{M} q(\nu_m) \prod_{n=1}^{M} q(\mathbf{m}_m) q(\mathbf{R}_m) \prod_{m=1}^{M} q(\mathbf{B}_m, \mathbf{\Sigma}_m)\, q(\theta_m)\,. \tag{4.60}$$

In the equation above and in the remaining of the paper, we introduce a maximum number M in the number of models. This can be seen as introducing for implementation reasons a truncation level in the stick-breaking representation of the underlying DP. It must be noted at this point, that M does not alter the

infinite mixture prior. It is just part of the variational approximation of the true posterior and required to make the underlying calculations feasible.

One now proceeds by iteratively maximizing Eq. (4.58) with respect to each one of the factors of Eq. (4.60) until a self-consistent solution is found. In particular, for any $\omega \in \mathbf{F}_\psi = \{z_1, \ldots, z_n, \nu_1, \ldots, \nu_M, \mathbf{m}_1, \ldots, \mathbf{m}_M, \mathbf{R}_1, \ldots, \mathbf{R}_M, (\mathbf{B}_1, \Sigma_1), \ldots, (\mathbf{B}_M, \Sigma_M), \theta_1, \ldots, \theta_M\}$, the update equation for $q(\omega)$ is given as:

$$\ln q(\omega) = \mathbb{E}_{\mathbf{F}_\psi \setminus \omega} \left[\ln p(\Psi, \mathcal{D}|\mathcal{I}) \right] + \text{ const} , \tag{4.61}$$

where the "const" denotes the normalization factor to the corresponding distribution, and the expectation is with respect to $q(\Psi)$ over all variables in $\mathbf{F}$ except ω. For a proof of Eq. (4.61), see Appendix C.3. All of these updates have an analytical form except the ones involving $\theta_m, m = 1, \ldots, M$. In what follows, we derive them one by one and also discuss how to perform the θ_m updates in an approximate non-parametric manner.

Update of $q(\nu_m)$

To compute the update of $q(\nu_m)$, we start with Eq. (4.61), use Eq. (4.59) and keep only the terms that depend on ν_m:

$$\boldsymbol{\ln q(\nu_m) = \ln p(\nu_m|\alpha_0) + \sum_{n=1}^{N} \mathbb{E}_{\mathbf{F}_\psi \setminus \nu_m}[\ln p(z_n|\mathbf{X}, \mathbf{m}, \mathbf{R}, \nu)] + \text{ const} ,} \tag{4.62}$$

where using Eqs. (4.36) and (4.44)

$$\begin{aligned}
& \mathbb{E}_{\mathbf{F}_\psi \setminus \nu_m}[\ln p(z_n|\mathbf{X}, \mathbf{m}, \mathbf{R}, \nu)] \\
= \; & \mathbb{E}_{\mathbf{F}_\psi \setminus \nu_m} \left[\ln \left(\prod_{i=1}^{M} \left(\frac{p(\mathbf{x}_n|\mathbf{m}_m, \mathbf{R}_m)}{\widehat{C}_n(\widehat{\mathbf{m}}, \widehat{\mathbf{R}}, \widehat{\nu})} \nu_i \prod_{j=1}^{i-1} \left(1 - \nu_j\right) \right)^{\mathbf{1}[z_n = i]} \right) \right] \\
= \; & \mathbb{E}_{\mathbf{F}_\psi \setminus \nu_m} \left[q(z_n = m) \ln \nu_m + \sum_{i=m+1}^{M} q(z_n = i) \ln (1 - \nu_m) \right] + \text{ const} \\
= \; & \mathbb{E}_{z_n}[q(z_n = m)] \ln \nu_m + \mathbb{E}_{z_n}[q(z_n > m)] \ln (1 - \nu_m) + \text{ const} .
\end{aligned}$$

Note here $\widehat{C}_n(\widehat{\mathbf{m}}, \widehat{\mathbf{R}}, \widehat{\nu})$ is calculated using the mean of the approximated posterior distribution from the previous iteration step, i.e. it is taken as constant that does not contribute to the update of $q(\nu_m)$.

Finally using Eq. (4.35):

$$\ln p\left(\nu_m|\alpha_0\right) = \left(\alpha_0 - 1\right) \ln\left(1 - \nu_m\right).$$

Thus, we can write:

$$\begin{aligned} & \ln q\left(\nu_m\right) \\ = \; & \left(\alpha_0 - 1\right) \ln\left(1 - \nu_m\right) + \sum_{n=1}^{N} \left\{\mathbb{E}_{z_n}[q(z_n = m)] \ln \nu_m + \mathbb{E}_{z_n}[q(z_n > m)] \ln\left(1 - \nu_m\right)\right\} + \text{ const} \\ = \; & \left[\sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n = m)]\right] \ln \nu_m + \left[\alpha_0 - 1 + \sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n > m)]\right] \ln\left(1 - \nu_m\right) + \text{ const}. \end{aligned}$$

From the above equation, we conclude that $q\left(\nu_m\right)$ follows a Beta distribution:

$$\nu_m \sim \mathcal{B}eta\left(1 + \sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n = m)], \alpha_0 + \sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n > m)]\right). \tag{4.63}$$

Update of $q(\mathbf{m}_m)$

To update of $q(\mathbf{m}_m)$, again, we use Eq. (4.61) and Eq. (4.59) and keep only the terms that depend on $\mathbf{m}_m$, so we have:

$$\mathbf{\ln q(m_m) = \ln p(m_m|u_0, R_0) + \sum_{n=1}^{N} \mathbb{E}_{F_\psi \setminus m_m}[\ln p(z_n|X, m, R, \nu)] + \text{ const}.} \tag{4.64}$$

The first prior term gives:

$$\ln p(\mathbf{m}_m|\mathbf{u}_0, \mathbf{R}_0) = -\frac{1}{2}(\mathbf{m}_m - \mathbf{u}_0)^T \mathbf{R}_0 (\mathbf{m}_m - \mathbf{u}_0) + \text{ const}. \tag{4.65}$$

The second term can be calculated as:

$$\mathbb{E}_{\mathbf{F}_\psi \setminus \mathbf{m}_m}[\ln p(z_n|\mathbf{X}, \mathbf{m}, \mathbf{R}, \nu)]$$

$$
\begin{aligned}
&= \mathbb{E}_{\mathbf{F}_\psi \setminus \mathbf{m}_m}\left[\ln\left(\prod_{i=1}^{M}\left(\frac{p(\mathbf{x}_n|\mathbf{m}_i,\mathbf{R}_i)\pi_i(\nu)}{\widehat{C}_n(\widehat{\mathbf{m}},\widehat{\mathbf{R}},\widehat{\nu})}\right)^{\mathbf{1}[z_n=i]}\right)\right] \\
&= \mathbb{E}_{\mathbf{F}_\psi \setminus \mathbf{m}_m}\left[\ln\left(\prod_{i=1}^{M} p(\mathbf{x}_n|\mathbf{m}_i,\mathbf{R}_i)^{\mathbf{1}[z_n=i]}\right)\right] + \text{ const} \\
&= \sum_{j=1}^{M}\left\{\mathbb{E}_{z_n}[q(z_n=i)]\mathbb{E}_{\mathbf{F}_\psi \setminus \mathbf{m}_m}\left[\ln p(\mathbf{x}_n|\mathbf{m}_i,\mathbf{R}_i)\right]\right\} \\
&= \mathbb{E}_{z_n}[q(z_n=m)]\mathbb{E}_{\mathbf{F}_\psi \setminus \mathbf{m}_m}\left[\ln p(\mathbf{x}_n|\mathbf{m}_m,\mathbf{R}_m)\right] \\
&= \mathbb{E}_{z_n}[q(z_n=m)]\left(-\frac{1}{2}(\mathbf{x}_n-\mathbf{m}_m)^T\mathbb{E}_{\mathbf{R}_m}[\mathbf{R}_m](\mathbf{x}_n-\mathbf{m}_m)\right) + \text{ const.}
\end{aligned}
$$

Substituting in Eq. (4.64), we obtain:

$$
\begin{aligned}
\ln q(\mathbf{m}_m) &= -\frac{1}{2}(\mathbf{m}_m-\mathbf{u}_0)^T\mathbf{R}_0(\mathbf{m}_m-\mathbf{u}_0) \\
&\quad -\frac{1}{2}\sum_{n=1}^{N}\left\{\mathbb{E}_{z_n}[q(z_n=m)](\mathbf{x}_n-\mathbf{m}_m)^T\mathbb{E}_{\mathbf{R}_m}[\mathbf{R}_m](\mathbf{x}_n-\mathbf{m}_m)\right\} + \text{ const} \\
&= -\frac{1}{2}\left(\mathbf{m}_m^T\mathbf{R}_0\mathbf{m}_m - 2\mathbf{m}_m^T\mathbf{R}_0\mathbf{u}_0 + \sum_{n=1}^{N}\mathbb{E}_{z_n}[q(z_n=m)]\mathbf{m}_m^T\mathbb{E}_{\mathbf{R}_m}[\mathbf{R}_m]\mathbf{m}_m\right. \\
&\quad \left. -2\sum_{n=1}^{N}\mathbb{E}_{z_n}[q(z_n=m)]\mathbf{m}_m^T\mathbb{E}_{\mathbf{R}_m}[\mathbf{R}_m]\mathbf{x}_n\right).
\end{aligned}
$$

For notational convenience, let us denote $\mathbf{R}_{m1} = \sum_{n=1}^{N}\mathbb{E}_{z_n}[q(z_n=m)]\mathbb{E}_{\mathbf{R}_m}[\mathbf{R}_m]$ and $\mathbf{R}_{m2} = \sum_{n=1}^{N}\mathbb{E}_{z_n}[q(z_n=m)]\mathbb{E}_{\mathbf{R}_m}[\mathbf{R}_m]\mathbf{x}_n$. It can easily be shown now that $q(\mathbf{m}_m)$ follows a Gaussian distribution:

$$
\mathbf{m}_m \sim \mathcal{N}_d\left(\mathbf{u}_m, (\mathbf{R}_0+\mathbf{R}_{m1})^{-1}\right), \tag{4.66}
$$

where $\mathbf{u}_m = (\mathbf{R}_0+\mathbf{R}_{m1})^{-1}(\mathbf{R}_0\mathbf{u}_0+\mathbf{R}_{m2})$.

Update of $q(\mathbf{R}_m)$

To update of $q(\mathbf{R}_m)$, we similarly keep only terms that are dependent on $\mathbf{R}_m$, thus obtaining:

$$\ln q(\mathbf{R}_m) = \ln p(\mathbf{R}_m|\mathbf{W}_0, \upsilon_0) + \sum_{n=1}^{N} \mathbb{E}_{\mathbf{F}_\psi \setminus \mathbf{R}_m}[\ln p(z_n|\mathbf{X}, \mathbf{m}, \mathbf{R}, \nu)] + \text{ const}. \tag{4.67}$$

The first prior term gives:

$$\ln p(\mathbf{R}_m|\mathbf{W}_0, \upsilon_0) = \frac{\upsilon_0 - d - 1}{2} \ln |\mathbf{R}_m| - \frac{1}{2} tr\left(\mathbf{R}_m \mathbf{W}_0^{-1}\right) + \text{ const}. \tag{4.68}$$

The expectation term can be calculated as:

$$\begin{aligned}
& \mathbb{E}_{\mathbf{F}_\psi \setminus \mathbf{R}_m}[\ln p(z_n|\mathbf{X}, \mathbf{m}, \mathbf{R}, \nu)] \\
= \; & \mathbb{E}_{\mathbf{F}_\psi \setminus \mathbf{R}_m}\left[\ln\left(\prod_{i=1}^{M}\left(\frac{p(\mathbf{x}_n|\mathbf{m}_i, \mathbf{R}_i)\pi_i(\nu)}{\widehat{C}_n(\widehat{\mathbf{m}}, \widehat{\mathbf{R}}, \widehat{\nu})}\right)^{\mathbf{1}[z_n=i]}\right)\right] \\
= \; & \mathbb{E}_{\mathbf{F}_\psi \setminus \mathbf{R}_m}\left[\ln\left(\prod_{i=1}^{M} p(\mathbf{x}_n|\mathbf{m}_i, \mathbf{R}_i)^{\mathbf{1}[z_n=i]}\right)\right] + \text{ const} \\
= \; & \mathbb{E}_{z_n}[q(z_n = m)]\mathbb{E}_{\mathbf{m}_m}\left[\ln p(\mathbf{x}_n|\mathbf{m}_m, \mathbf{R}_m)\right] + \text{ const} \\
= \; & \frac{1}{2}\mathbb{E}_{z_n}[q(z_n = m)]\left(\ln |\mathbf{R}_m| - \mathbb{E}_{\mathbf{m}_m}\left[(\mathbf{x}_n - \mathbf{m}_m)^T \mathbf{R}_m (\mathbf{x}_n - \mathbf{m}_m)\right]\right) + \text{ const}.
\end{aligned}$$

Since $(\mathbf{x}_n - \mathbf{m}_m)^T \mathbf{R}_m (\mathbf{x}_n - \mathbf{m}_m)$ is a scalar, we can write:

$$tr\left[(\mathbf{x}_n - \mathbf{m}_m)^T \mathbf{R}_m (\mathbf{x}_n - \mathbf{m}_m)\right] = tr\left[\mathbf{R}_m (\mathbf{x}_n - \mathbf{m}_m)(\mathbf{x}_n - \mathbf{m}_m)^T\right], \tag{4.69}$$

and thus simplify as:

$$\begin{aligned}
& \mathbb{E}_{\mathbf{F}_\psi \setminus \mathbf{R}_m}[\ln p(\mathbf{x}_n|z_n, \mathbf{m}_{z_n}, \mathbf{R}_{z_n})] \\
= \; & \frac{1}{2}\mathbb{E}_{z_n}[q(z_n = m)]\left(\ln |\mathbf{R}_m| - tr\left(\mathbf{R}_m \mathbb{E}_{\mathbf{m}_m}\left[(\mathbf{x}_n - \mathbf{m}_m)(\mathbf{x}_n - \mathbf{m}_m)^T\right]\right)\right) + \text{ const}.
\end{aligned}$$

Finally, we can write:

$$\ln q(\mathbf{R}_m)$$

$$= \frac{\sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n = m)] + \upsilon_0 - d - 1}{2} \ln |\mathbf{R}_m| \\ -\frac{1}{2} tr\left(\mathbf{R}_m \left(\mathbf{W}_0^{-1} + \sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n = m)] \mathbb{E}_{\mathbf{m}_m}\left[(\mathbf{x}_n - \mathbf{m}_m)(\mathbf{x}_n - \mathbf{m}_m)^T\right]\right)\right) + \text{ const.}$$

Thus $\mathbf{R}_m$ follows a Wishart distribution:

$$\mathbf{R}_m \sim \mathcal{W}_d(\mathbf{W}_m, \upsilon_m), \tag{4.70}$$

where

$$\upsilon_m = \sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n = m)] + \upsilon_0, \tag{4.71}$$

$$\mathbf{W}_m^{-1} = \mathbf{W}_0^{-1} + \sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n = m)] \mathbb{E}_{\mathbf{m}_m}\left[(\mathbf{x}_n - \mathbf{m}_m)(\mathbf{x}_n - \mathbf{m}_m)^T\right]. \tag{4.72}$$

Update of $q(z_n)$

Similarly to find the update of $q(z_n)$, we start with Eq. (4.61), use Eq. (4.59) and keep only the terms that depend on z_n:

$$\begin{aligned} \ln q(z_n) &= \mathbb{E}_{\mathbf{F}_\psi \setminus z_n}\left[\ln p(z_n|\mathbf{X}, \mathbf{m}, \mathbf{R}, \nu) + \ln p(\mathbf{y}_n|\mathbf{x}_n, z_n, \Phi, \mathcal{D})\right] + \text{ const} \\ &= \mathbb{E}_{\mathbf{F}_\psi \setminus z_n}\left[\ln\left(\prod_{m=1}^{M}\left(\frac{p(\mathbf{x}_n|\mathbf{m}_m, \mathbf{R}_m)\pi_m(\nu)}{\widehat{C}_n(\widehat{\mathbf{m}}, \widehat{\mathbf{R}}, \widehat{\nu})}\right)^{\mathbf{1}[z_n=m]}\right) + \ln p(\mathbf{y}_n|\mathbf{x}_n, z_n, \Phi, \mathcal{D})\right] + \text{ const.} \end{aligned} \tag{4.73}$$

The first term of the right-hand side of the above equation using Eqs. (4.36) and (4.44) gives the following:

$$\mathbb{E}_{\mathbf{F}_\psi \setminus z_n}\left[\ln\left(\prod_{m=1}^{M}\left(\frac{p(\mathbf{x}_n|\mathbf{m}_m, \mathbf{R}_m)\pi_m(\nu)}{\widehat{C}_n(\widehat{\mathbf{m}}, \widehat{\mathbf{R}}, \widehat{\nu})}\right)^{\mathbf{1}[z_n=m]}\right)\right] \\ = \sum_{m=1}^{M}\left\{\mathbf{1}[z_n = m]\left(\mathbb{E}_\nu[\pi_m(\nu)] + \mathbb{E}_{\mathbf{F}_\psi \setminus z_n}[\ln p(\mathbf{x}_n|\mathbf{m}_m, \mathbf{R}_m)] - \ln\left(\widehat{C}_n(\widehat{\mathbf{m}}, \widehat{\mathbf{R}}, \widehat{\nu})\right)\right)\right\}$$

$$= \sum_{m=1}^{M} \left\{ \mathbf{1}[z_n = m] \left(\mathbb{E}_{\nu_m}[\ln \nu_m] + \sum_{j=1}^{m-1} \mathbb{E}_{\nu_j}\left[\ln(1-\nu_j)\right] + \frac{1}{2}\mathbb{E}_{\mathbf{R}_m}[\mathbf{R}_m] \right.\right.$$
$$\left.\left. -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\mathbb{E}_{\mathbf{m}_m,\mathbf{R}_m}\left[(\mathbf{x}_n - \mathbf{m}_m)^T \mathbf{R}_m (\mathbf{x}_n - \mathbf{m}_m)\right] - \ln\left(\widehat{C}_n(\widehat{\mathbf{m}}, \widehat{\mathbf{R}}, \widehat{\nu})\right) \right) \right\}, \quad (4.74)$$

where $\mathbb{E}_{\mathbf{m}_m,\mathbf{R}_m}\left[(\mathbf{x}_n - \mathbf{m}_m)^T \mathbf{R}_m (\mathbf{x}_n - \mathbf{m}_m)\right]$ is calculated as [14]:

$$\mathbb{E}_{\mathbf{m}_m,\mathbf{R}_m}\left[(\mathbf{x}_n - \mathbf{m}_m)^T \mathbf{R}_m (\mathbf{x}_n - \mathbf{m}_m)\right] = \upsilon_m (\mathbf{x}_n - \mathbf{u}_m)^T \mathbf{W}_m (\mathbf{x}_n - \mathbf{u}_m). \quad (4.75)$$

Here $\mathbf{u}_m$, υ_m, and $\mathbf{W}_m$ are given by Eqs. (4.66) and (4.70). Note here that even though $\widehat{C}_n(\widehat{\mathbf{m}}, \widehat{\mathbf{R}}, \widehat{\nu})$ is taken as a constant, it still contributes to the update of $q(\mathbf{z})$.

For computing the second term of Eq. (4.73), we consider the predictive distribution with $\mathbf{B}$ and Σ integrated out (Eq. (4.23)):

$$p(\mathbf{y}_n|\mathbf{x}_n, z_n = m, \theta_m, \mathcal{D}_m) = \mathcal{T}_q(\mu^{**}(\mathbf{x}_n), c^{**}(\mathbf{x}_n, \mathbf{x}_n; \theta_m)\widehat{\Sigma}_m; n_m - p),$$
$$= c_m |\Lambda_m|^{-1/2} \left| \mathbb{I}_q + \frac{1}{n_m - p} (\mathbf{y}_n - \mu^{**}(\mathbf{x}_n))^T \Lambda_m^{-1} (\mathbf{y}_n - \mu^{**}(\mathbf{x}_n)) \right|^{-(n_m - p + q)/2},$$

where we defined $c_m = (\pi(n_m - p))^{-q/2} \frac{\Gamma[(n_m - p + q)/2]}{\Gamma[(n_m - p)/2]}$ and $\boldsymbol{\Lambda_m = c^{**}(x_n, x_n; \theta_m)\widehat{\Sigma}_m}$.

We can now compute the third term of the right-hand side of Eq. (4.73) as:

$$\mathbb{E}_{\mathbf{F}_\psi \setminus z_n}[\ln p(\mathbf{y}_n|\mathbf{x}_n, z_n, \Theta, \mathcal{D})]$$
$$= \sum_{m=1}^{M} \left\{ \mathbf{1}[z_n = m] \mathbb{E}_{\mathbf{F}_\psi \setminus z_n}\left[\ln p(\mathbf{y}_n|z_n = m, \mathbf{x}_n, \theta_m, \mathcal{D}_m)\right] \right\}$$
$$= \sum_{m=1}^{M} \left\{ \mathbf{1}[z_n = m] \left(\ln c_m - \frac{1}{2}\mathbb{E}_{\theta_m}[\ln |\Lambda_m|] \right.\right.$$
$$\left.\left. -\frac{n_m - p + q}{2} \mathbb{E}_{\theta_m}\left[\ln \left| \mathbb{I}_q + \frac{1}{n_m - p} (\mathbf{y}_n - \mu^{**}(\mathbf{x}_n))^T \Lambda_m^{-1} (\mathbf{y}_n - \mu^{**}(\mathbf{x}_n)) \right| \right] \right) \right\}. \quad (4.76)$$

Remark 4. The terms above that involve the expectation, $\mathbb{E}_{\theta_m}[\cdot]$, are computed as follows. We first approximate $q(\theta_m)$ by a Gaussian mixture as

$$\boldsymbol{q(\theta_m) = \frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{l=1}^{L} \omega_l^2 \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2 \mathbb{I}_{d+1}).} \quad (4.77)$$

Let us denote the function inside of the expectation operator as $h(\theta_m)$. We can approximate $h(\theta_m)$ around each center of the Gaussian mixture $\mathbf{m}_l$ using a first-order Taylor series expansion as follows [38]:

$$\boldsymbol{h(\theta_m) \approx \hat{h}_l(\theta_m) = h(\mathbf{m}_l) + \nabla h(\mathbf{m}_l)(\theta_m - \mathbf{m}_l).} \tag{4.78}$$

Then $\mathbb{E}_{\theta_m}[h(\theta_m)]$ can be approximated as

$$\begin{aligned}\mathbb{E}_{\theta_m}[h(\theta_m)] &\approx \frac{1}{\sum_{l=1}^{L}\omega_l^2}\sum_{j=1}^{L}\int_{\theta_m}\omega_l^2\mathcal{N}(\theta_m;\mathbf{m}_l,\sigma_l^2\mathbb{I}_{d+1})\hat{h}_l(\theta_m)d\theta_m \\ &= \frac{1}{\sum_{l=1}^{L}\omega_l^2}\sum_{j=1}^{L}\omega_l^2 h(\mathbf{m}_l).\end{aligned}$$

Note that the first derivative of $h(\theta_m)$, $\nabla h(\theta_m)$ does not enter this calculation. However, if a second-order Taylor expansion with respect to θ_m is used instead of Eq. (4.78), one would need to compute $\nabla^2 h(\mathbf{m}_l)$.

Combining the terms in Eqs. (4.74) and (4.76), the variational update for z_n is given by

$$q(z_n) = \prod_{m=1}^{M}[\widehat{\rho}_{n,m}]^{\mathbf{1}[z_n=m]}. \tag{4.79}$$

Here, $\widehat{\rho}_{n,m} = \left(\frac{\rho_{n,m}}{\sum_{m=1}^{M}\rho_{n,m}}\right)$ is the *normalized responsibility*, and $\rho_{n,m}$ is given as:

$$\begin{aligned}\rho_{n,m} &= \mathbb{E}_{\nu_m}[\ln\nu_m] + \sum_{j=1}^{m-1}\mathbb{E}_{\nu_j}\left[\ln(1-\nu_j)\right] + \frac{1}{2}\mathbb{E}_{\mathbf{R}_m}[\mathbf{R}_m] - \frac{d}{2}\ln(2\pi) \\ &\quad -\frac{1}{2}\mathbb{E}_{\mathbf{m}_m,\mathbf{R}_m}\left[(\mathbf{x}_n-\mathbf{m}_m)^T\mathbf{R}_m(\mathbf{x}_n-\mathbf{m}_m)\right] - \ln\left(\widehat{C}_n(\widehat{\mathbf{m}},\widehat{\mathbf{R}},\widehat{\nu})\right) + \ln c_m - \frac{1}{2}\mathbb{E}_{\theta_m}\left[\ln|\Lambda_m|\right] \\ &\quad -\frac{n_m-p+q}{2}\mathbb{E}_{\theta_m}\left[\ln\left|\mathbb{I}_q + \frac{1}{n_m-p}(\mathbf{y}_n-\mu^{**}(\mathbf{x}_n))^T\Lambda_m^{-1}(\mathbf{y}_n-\mu^{**}(\mathbf{x}_n))\right|\right].\end{aligned}$$

Remark 5. Upon convergence of the VI algorithm and computation of the responsibilities $\widehat{\rho}_{n,m}$, the classification of the observation data $\mathcal{D}$ is based on the maximum responsibilities. For example, we assign $(\mathbf{x}_n, \mathbf{y}_n)$ to the $i-$th model if

$\text{argmax}_m \widehat{\rho}_{n,m} = i$, i.e., each mixture component is taken to completely model a subset of the observation data. The clustering process is similar to the works in [95] and [15] and simplifies the training of the algorithm by having each model explaining only a small portion of the data set.

Remark 6. The number of data points assigned to each model has to be larger than $p + q$ in order to ensure a proper posterior distribution for the covariance matrix Σ_m (see also Section 4.1.4). Therefore, any model that is assigned less than $p + q$ points is removed and the corresponding data are re-assigned to the models with the next greatest responsibility in explaining them.

Update of $q(\mathbf{B}_m, \Sigma_m)$

Once our data $\mathcal{D}$ are classified using $q(\mathbf{z})$, we proceed to update independently the hyper-parameters of each model. Let us consider the m-th model as an example. We first discuss the update of $q(\mathbf{B}_m, \Sigma_m)$, and then proceed to the update of $q(\theta_m)$.

From the results of Appendix C.1 (Eq. (C.3)), we can directly write the update of $q(\mathbf{B}_m|\Sigma_m)$ as:

$$\boldsymbol{q(\mathbf{B}_m|\Sigma_m) = \mathcal{N}_{p\times q}\left(\mathbf{B}_m|\widehat{\mathbf{B}}_m, \mathbb{E}_{\theta_m}\left[(\mathbf{H}_m^T\mathbf{A}_m^{-1}\mathbf{H}_m)^{-1}\right], \Sigma_m\right),} \tag{4.80}$$

where from Eq. (4.26)

$$\widehat{\mathbf{B}}_m = \mathbb{E}_{\theta_m}\left[(\mathbf{H}_m^T\mathbf{A}_m^{-1}\mathbf{H}_m)^{-1}\mathbf{H}_m^T\mathbf{A}_m^{-1}\mathbf{Y}_m\right]. \tag{4.81}$$

Similarly, from Appendix C.1 (Eq. (C.5)), we can write the update of $q(\Sigma_m)$

as:

$$q(\boldsymbol{\Sigma}_m) = \mathcal{W}_q^{-1}(\boldsymbol{\Sigma}_m|(n_m - p)\widehat{\boldsymbol{\Sigma}}_m, n_m - p), \tag{4.82}$$

where from Eq. (C.6), we define: $\widehat{\Sigma}_m = \frac{1}{n_m - p}\mathbb{E}_{\theta_m}\left[(\mathbf{Y}_m - \mathbf{H}_m\widehat{\mathbf{B}}_m)^T\mathbf{A}_m^{-1}(\mathbf{Y}_m - \mathbf{H}_m\widehat{\mathbf{B}}_m)\right]$.

The expectation values are approximated here using the first-order Taylor series approximation discussed earlier in Remark 4.

Remark 7**.** The updates above require the inversion of the local covariance matrices. This scales as the cube of the number of data points assigned to a cluster. For a single MGP, the scaling is $O(N^3)$ where N is the size of the data set. For a M-mixture MGP model, the average scaling is $O(Mm^3)$ where m is the average number of points per cluster. Assuming that $m \sim N/M$, one obtains a scaling of $O(N^3/M^2)$ which is much better than $O(N^3)$ (with one cluster). Thus the mixture model scales much better than using a single MGP. In addition, these calculations can be done in parallel and fast especially if the separable covariance function approach in [11] is exploited.

Update of $q(\theta_m)$

The update of $q(\theta_m)$ is slightly more complex. This is due to the fact that the target posterior $p(\theta_m|\mathbf{B}_m, \Sigma_m, \mathcal{D}_m)$ and thus $q(\theta_m)$ cannot be represented by any of the standard distributions (see Appendix C.1). A recently developed nonparametric variational inference algorithm [38] provides us a nonparametric approach for approximating $q(\theta_m)$ via Gaussian mixtures.

Let us denote the m-th data subset as $\mathcal{D}_m$. The lower bound of the local

evidence $\ln p(\mathcal{D}_m)$ can be expressed as:

$$\boldsymbol{\mathcal{L}[q] = \mathbb{E}_q\left[\ln \frac{p(\theta_m, \mathcal{D}_m)}{q(\theta_m)}\right] = \mathcal{H}[q] + \mathbb{E}_q[g(\theta_m)],} \tag{4.83}$$

where $\mathcal{H}[q]$ is the entropy of $q(\theta_m)$,

$$\begin{aligned} g(\theta_m) &= \mathbb{E}_{\Psi\setminus\theta_m}\left[\ln \pi\left(\theta_m|\gamma\right) + \ln p\left(\mathcal{S}_m\left(\mathbf{Y};\mathbf{z}\right)|\mathcal{S}_m\left(\mathbf{X};\mathbf{z}\right), \phi_m\right)\right] + \text{ const} \\ &= \ln \pi\left(\theta_m|\gamma\right) + \mathbb{E}_{\mathbf{B}_m,\Sigma_m}\left[\ln p\left(\mathcal{S}_m\left(\mathbf{Y};\mathbf{z}\right)|\mathcal{S}_m\left(\mathbf{X};\mathbf{z}\right), \theta_m, \mathbf{B}_m, \Sigma_m\right)\right] + \text{ const}, \end{aligned} \tag{4.84}$$

and using Eq. (4.15) (also Eq. (C.1)):

$$\begin{aligned} & \mathbb{E}_{\mathbf{B}_m,\Sigma_m}\left[\ln p\left(\mathcal{S}_m\left(\mathbf{Y};\mathbf{z}\right)|\mathcal{S}_m\left(\mathbf{X};\mathbf{z}\right), \theta_m, \mathbf{B}_m, \Sigma_m\right)\right] \\ = \; & \mathbb{E}_{\mathbf{B}_m,\Sigma_m}\left[\ln \frac{\exp\left(-\frac{1}{2}tr\left[\Sigma_m^{-1}\left(\mathcal{S}_m\left(\mathbf{Y};\mathbf{z}\right) - \mathbf{H}_m\mathbf{B}_m\right]\right)^T \mathbf{A}_m^{-1}\left(\mathcal{S}_m\left(\mathbf{Y};\mathbf{z}\right) - \mathbf{H}_m\mathbf{B}_m\right)\right)}{(2\pi)^{n_m q/2}|\Sigma_m|^{n_m/2}|\mathbf{A}_m|^{q/2}}\right] \\ \approx \; & -\frac{1}{2}tr\left[\mathbb{E}[\Sigma_m^{-1}]\left(\mathcal{S}_m\left(\mathbf{Y};\mathbf{z}\right) - \mathbf{H}_m\mathbb{E}[\mathbf{B}_m]\right)^T \mathbf{A}_m^{-1}\left(\mathcal{S}_m\left(\mathbf{Y};\mathbf{z}\right) - \mathbf{H}_m\mathbb{E}[\mathbf{B}_m]\right)\right] \\ & - \ln\left((2\pi)^{n_m q/2}\mathbb{E}[|\Sigma_m|]^{n_m/2}|\mathbf{A}_m|^{q/2}\right). \end{aligned} \tag{4.85}$$

Following [38] (see also Remark 4, we choose the distribution of $q(\theta_m)$ to be a weighted Gaussian mixture with isotropic covariance as in Eq. (4.77).

Next, we compute the lower bound of the entropy term $\mathcal{H}[q]$, and then maximize the new lower bound of the local evidence. The lower bound of the entropy can be found using Jensen's inequality [52],

$$\begin{aligned} \mathcal{H}[q] &= -\int_{\theta_m} q(\theta_m)\ln q(\theta_m)d\theta_m \\ &= -\int_{\theta_m} \frac{1}{\sum_{l=1}^L \omega_l^2}\sum_{l=1}^L \omega_l^2 \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2\mathbb{I}_{d+1})\ln q(\theta_m)d\theta_m \\ &= -\frac{1}{\sum_{l=1}^L \omega_l^2}\sum_{l=1}^L \omega_l^2 \int_{\theta_m} \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2\mathbb{I}_{d+1})\ln q(\theta_m)d\theta_m \\ &\geq -\frac{1}{\left(\sum_{l=1}^L \omega_l^2\right)^2}\sum_{l=1}^L \omega_l^2 \ln q_l, \end{aligned} \tag{4.86}$$

where $q_l = \sum_{j=1}^{L} \omega_j^2 q'_{lj}$, $q'_{lj} = \mathcal{N}(\mathbf{m}_l; \mathbf{m}_j, (\sigma_l^2 + \sigma_j^2)\mathbb{I}_{d+1})$. The proof can be found in Appendix C.4.

We now look at the expected log local joint $g(\theta_m)$,

$$\mathbb{E}_q[g(\theta_m)] = \frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{j=1}^{L} \omega_l^2 \int_{\theta_m} \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2 \mathbb{I}_{d+1}) g(\theta_m) d\theta_m. \tag{4.87}$$

The local joint $g(\theta_m)$ can be approximated around each site of the proposals $\mathbf{m}_l$ with a second-order Taylor series expansion,

$$g(\theta_m) \approx \hat{g}_l(\theta_m) = g(\mathbf{m}_l) + \nabla g(\mathbf{m}_l)(\theta_m - \mathbf{m}_l) + \frac{1}{2}(\theta_m - \mathbf{m}_l)^T \mathcal{H}_l (\theta_m - \mathbf{m}_l), \tag{4.88}$$

where $\mathcal{H}_l = \nabla^2_{\theta_m} g(\theta_m)$ is the Hessian matrix. The approximate expectation can be then written as:

$$\begin{aligned} \mathbb{E}_q[g(\theta_m)] &\approx \frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{j=1}^{L} \int_{\theta_m} \omega_l^2 \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2 \mathbb{I}_{d+1}) \hat{g}_l(\theta_m) d\theta_m. \\ &= \frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{j=1}^{L} \omega_l^2 \left\{ g(\mathbf{m}_l) + \frac{\sigma_l^2}{2} tr(\mathcal{H}_l) \right\}. \end{aligned} \tag{4.89}$$

The proof can be found in Appendix C.5. Finally, combined with the bound of the entropy term in Eq. (4.86), we derive the following approximation of the lower bound of the local evidence,

$$\mathcal{L}_2[q(\theta)] = \frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{j=1}^{L} \omega_l^2 \left\{ g(\mathbf{m}_l) + \frac{\sigma_l^2}{2} tr(\mathcal{H}_l) \right\} - \frac{1}{(\sum_{l=1}^{L} \omega_l^2)^2} \sum_{l=1}^{L} \omega_l^2 \ln q_l. \tag{4.90}$$

The variational parameters ω_l, $\mathbf{m}_l$ and σ_l are learnt by a gradient ascent method. Following [38], in the update of $\mathbf{m}_l$, we only use the first-order approximated lower bound of the local evidence to avoid the calculation of the gradient of the Hessian trace $tr(\mathcal{H}_l)$. The first-order approximation is given by

$$\mathcal{L}_1[q(\theta_m)] = \frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{j=1}^{L} \omega_l^2 g(\mathbf{m}_l) - \frac{1}{(\sum_{l=1}^{L} \omega_l^2)^2} \sum_{l=1}^{L} \omega_l^2 \ln q_l. \tag{4.91}$$

The nonparametric variational inference procedure is summarized in Algorithm 5 (η_1, η_2, η_3 appropriate learning rates). The final forms of the needed derivatives $\frac{\partial \mathcal{L}_1[q]}{\partial \mathbf{m}_l}$, $\frac{\partial \mathcal{L}_2[q]}{\partial \sigma_l}$ and $\frac{\partial \mathcal{L}_2[q]}{\partial \omega_l}$ are given in Appendix C.6.

Algorithm 5: Nonparametric variational inference for MGP model

---

Input: training data set $\mathcal{D}_m$, number of kernels L.

Initialize: $\{\omega_l, \mathbf{m}_l, \sigma_l\}_{l=1}^{L}$ **randomly. Set** $t = 0$**.**

repeat

for $l = 1, \ldots, L$ **do**

$(\mathbf{m}_l)^{(t+1)} = (\mathbf{m}_l)^{(t)} + \eta_1 \frac{\partial \mathcal{L}_1[q]}{\partial \mathbf{m}_l}$**.**

end for

for $l = 1, \ldots, L$ **do**

$(\sigma_l)^{(t+1)} = (\sigma_l)^{(t)} + \eta_2 \frac{\partial \mathcal{L}_2[q]}{\partial \sigma_l}$**.**

end for

for $l = 1, \ldots, L$ **do**

$(\omega_l)^{(t+1)} = (\omega_l)^{(t)} + \eta_3 \frac{\partial \mathcal{L}_2[q]}{\partial \omega_l}$**.**

end for

set $t = t + 1$**.**

until change of $\mathcal{L}_2[q(\theta_m)]$ **is less than** δ**.**

Finally, with ς_1, ς_2 given tolerances, we summarize the overall variational inference approach in Algorithm 6.

Algorithm 6: Computing the approximation $q(\Psi)$ to the posterior $p(\Psi|\mathcal{D}, \gamma, \alpha_0)$

Given: $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$, α_0, γ_r, γ_ϵ and tolerances ς_1, ς_2, and δ.

Set $t_1 = 0$ and the truncation level M

Initialize $q(\Psi)$ using the prior distribution

Randomly set $q^0(\mathbf{Z})$ by sampling from the prior distribution of the DP using Eq. (4.44)

while $|\mathcal{L}^{t_1+1}(q, \mathcal{D}) - \mathcal{L}^{t_1}(q, \mathcal{D})| < \varsigma_1$ **do**

Cluster the observations based on $q^{t_1}(\mathbf{Z})$ as discussed in Section 4.1.4. Let M_r be the number of remaining clusters.

for $m = 1, \ldots, M_r$ **do**

Update of $q^{t_1+1}(\mathbf{m}_m)$ using Eq. (4.66)

Update of $q^{t_1+1}(\mathbf{R}_m)$ using Eq. (4.70)

Set $t_2 = 0$

while $|\mathcal{L}_2^{t_2+1}(q_m, \mathcal{D}_m) - \mathcal{L}_2^{t_2}(q_m, \mathcal{D}_m)| < \varsigma_2$ **do**

Update of $q^{t_2+1}(\mathbf{B}_m|\Sigma_m)$ using Eq. (4.80)

Update of $q^{t_2+1}(\Sigma_m)$ using Eq. (4.82)

Update $q^{t_2+1}(\theta_m)$ using Algorithm 5 in Section 4.1.4.

Set $t_2 = t_2 + 1$.

end while

end for

Update $q^{t_1+1}(v)$ by maximizing $\mathcal{L}^{t_1}(q(v), \mathcal{D})$ using Eq. (4.63).

Update $q^{t_1+1}(\mathbf{z})$ by maximizing $\mathcal{L}^{t_1}(q(\mathbf{z}), \mathcal{D})$ using Eq. (4.79).

Calculate $\widehat{C}_n(\widehat{\mathbf{m}}, \widehat{\mathbf{R}}, \widehat{\nu})$ in Eq. (4.59) for all observations.

Set $t_1 = t_1 + 1$.

end while

4.1.5 Application to uncertainty quantification

As discussed in Chapter 1, the problem of UP is to propagate the uncertainty of the random variables ξ through $\mathbf{f}(\cdot)$ and characterize the statistics of the response such as the mean (see Eq. (4.4)), the covariance (see Eq. (4.5)), the PDF of the output at a particular location and time (see Eq. (4.6)), and so on. We will exploit the Bayesian nature of the surrogate we built to derive error bars for any computed statistic. These error bars correspond to the epistemic uncertainty induced by the limited amount of data in $\mathcal{D}$ which is, in turn, captured by the approximate posterior over the parameters $q(\Psi)$ of Eq. (4.60). The key idea is that every sample from $q(\Psi)$ yields a sample response surface $\widehat{\mathbf{f}}(\cdot; \Psi, \mathcal{D})$ which may be interrogated for the statistics. Mathematically, let $Q[\cdot]$ be a statistic of interest, i.e. a functional of the response surface. Then, we can obtain a probabilistic estimate for this statistic as follows:

$$p(Q|\mathcal{D}) = \int q(\Psi)\delta\left(Q\left[\widehat{\mathbf{f}}(\cdot; \Psi, \mathcal{D})\right] - Q\right) d\Psi. \tag{4.92}$$

The uncertainty in $p(Q|\mathcal{D})$ is due to the limited number of observations contained in $\mathcal{D}$.

The predictive distribution

In general, the prediction is desired to be made on a denser spatial points $\mathbf{S}^* \in \mathbb{R}^{n_s \times d_s}$ and/or time steps $\mathbf{t}^* \in \mathbb{R}^{n_t}$. Let ξ^* be one sample from the known distribution $p(\xi)$, $\mathbf{s}^* \in \mathbf{S}^*$, and $t^* \in \mathbf{t}^*$, and we denote $\mathbf{y}^*$ as the corresponding unknown output at the point $\mathbf{x}^* = (\xi^*, \mathbf{s}^*, t^*)$.

Then, the joint distribution of $\mathbf{y}^*$ and the hidden variables z^* given the new

input $\mathbf{x}^*$, all the observations and the parameters Ψ can be written as:

$$\begin{aligned} p(\mathbf{y}^*, z^*|\mathbf{x}^*, \Psi, \mathcal{D}) &= p(\mathbf{y}^*|\mathbf{x}^*, z^*, \mathbf{z}, \Phi, \mathcal{D}) p(z^*|\mathbf{x}^*, \mathbf{m}, \mathbf{R}, \nu) \\ &= p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{S}_{z^*}(\mathbf{X}; \mathbf{z}), \mathcal{S}_{z^*}(\mathbf{Y}; \mathbf{z}), \phi_{z^*}) p(z^*|\mathbf{x}^*, \mathbf{m}, \mathbf{R}, \nu). \end{aligned}$$

The first term of the right hand-side of the above equation is the predictive distribution of the z^*-th latent MGP given by Eq. (4.20). The second term represents the predictive responsibility that is given by Eq. (4.45):

$$p(z^*|\mathbf{x}^*, \mathbf{m}, \mathbf{R}, \nu) = \frac{p(\mathbf{x}^*|z^*, \mathbf{m}_{z^*}, \mathbf{R}_{z^*}) p(z^*|\nu)}{\sum_{\tilde{z}^*} p(\mathbf{x}^*|\tilde{z}^*, \mathbf{m}_{\tilde{z}^*}, \mathbf{R}_{\tilde{z}^*}) p(\tilde{z}^*|\nu)},$$

where $p(\mathbf{x}^*|z^*, \mathbf{m}_{z^*}, \mathbf{R}_{z^*})$ is given by Eq. (4.46), $p(z^*|\nu)$ is given by Eq. (4.44).

We may first multiply the joint distribution with the approximate posteriors $q(\mathbf{B}_m, \Sigma_m)$, $q(\mathbf{m})$, $q(\mathbf{R})$, and $q(\nu)$, and integrate them out. We have:

$$\begin{aligned} & p(\mathbf{y}^*, z^*|\mathbf{x}^*, \mathbf{z}, \Theta, \mathcal{D}) \\ = & \int p(\mathbf{y}^*|\mathbf{x}^*, z^*, \mathbf{z}, \Phi, \mathcal{D}) p(z^*|\mathbf{x}^*, \mathbf{m}, \mathbf{R}, \nu) q(\mathbf{B}_{z^*}, \Sigma_{z^*}) q(\mathbf{m}) q(\mathbf{R}) q(\nu) d\mathbf{B}_{z^*} d\Sigma_{z^*} d\mathbf{m} d\mathbf{R} d\nu. \\ = & \int p(\mathbf{y}^*|\mathbf{x}^*, z^*, \mathbf{z}, \Phi, \mathcal{D}) q(\mathbf{B}_{z^*}, \Sigma_{z^*}) d\mathbf{B}_{z^*} d\Sigma_{z^*} \int p(z^*|\mathbf{x}^*, \mathbf{m}, \mathbf{R}, \nu) q(\mathbf{m}) q(\mathbf{R}) q(\nu) d\mathbf{m} d\mathbf{R} d\nu. \end{aligned}$$

Here recall $\Theta = \{\theta_1, \ldots, \theta_M\}$.

The first integral term can be calculated as

$$\begin{aligned} & p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{S}_{z^*}(\mathbf{X}; \mathbf{z}), \mathcal{S}_{z^*}(\mathbf{Y}; \mathbf{z}), \theta_{z^*}) \\ = & \int p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{S}_{z^*}(\mathbf{X}; \mathbf{z}), \mathcal{S}_{z^*}(\mathbf{Y}; \mathbf{z}), \phi_{z^*}) q(\mathbf{B}_{z^*}, \Sigma_{z^*}) d\mathbf{B}_{z^*} d\Sigma_{z^*} \\ = & p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{S}_{z^*}(\mathbf{X}; \mathbf{z}), \mathcal{S}_{z^*}(\mathbf{Y}; \mathbf{z}), \theta_{z^*}, \widehat{\mathbf{B}}_{z^*}, \widehat{\Sigma}_{z^*}), \end{aligned}$$

where $\widehat{\mathbf{B}}_{z^*}$ and $\widehat{\Sigma}_{z^*}$ are the posterior mean given by Eq. (4.80) and Eq. (4.82), respectively, and they are depending on θ_{z^*}.

The second integral term is analytically intractable, so we simply approximate it using the mean of $q(\mathbf{m}), q(\mathbf{R})$ and $q(\nu)$. Mathematically, we take $q(\mathbf{m}) =$

$\delta(\widehat{\mathbf{m}} - \mathbf{m})$, where $\widehat{\mathbf{m}}$ is the posterior mean of $\mathbf{m}$. For $\mathbf{R}$ and ν, we do the similar trick. Then, we can write

$$\begin{aligned} p(z^*|\mathbf{x}^*) &= \int \frac{p(\mathbf{x}^*|z^*, \mathbf{m}_{z^*}, \mathbf{R}_{z^*})p(z^*|\nu)}{\sum_{\tilde{z}^*} p(\mathbf{x}^*|\tilde{z}^*, \mathbf{m}_{\tilde{z}^*}, \mathbf{R}_{\tilde{z}^*})p(\tilde{z}^*|\nu)} q(\mathbf{m})q(\mathbf{R})q(\nu)d\mathbf{m}d\mathbf{R}d\nu \\ &\approx \frac{p(\mathbf{x}^*|z^*, \widehat{\mathbf{m}}_{z^*}, \widehat{\mathbf{R}}_{z^*})p(z^*|\widehat{\nu})}{\sum_{\tilde{z}^*} p(\mathbf{x}^*|\tilde{z}^*, \widehat{\mathbf{m}}_{\tilde{z}^*}, \widehat{\mathbf{R}}_{\tilde{z}^*})p(\tilde{z}^*|\widehat{\nu})}, \end{aligned}$$

Now the joint distribution $p(\mathbf{y}^*, z^*|\mathbf{x}^*, \Psi, \mathcal{D})$ can be simplified as

$$p(\mathbf{y}^*, z^*|\mathbf{x}^*, \mathbf{z}, \Theta, \mathcal{D}) = p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{S}_{z^*}(\mathbf{X}; \mathbf{z}), \mathcal{S}_{z^*}(\mathbf{Y}; \mathbf{z}), \theta_{z^*})p(z^*|\mathbf{x}^*).$$

We can further multiply $q(\mathbf{z})$ and integrate $\mathbf{z}$ out. Here, as discussed in Remark 5, we choose $\widehat{\mathbf{z}}$ to be the MAP estimate of $\mathbf{z}$ as

$$\widehat{\mathbf{z}} = \arg\max_{\mathbf{z}} q(\mathbf{z}). \tag{4.93}$$

Then, we take $q(\mathbf{z}) \approx \delta(\widehat{\mathbf{z}} - \mathbf{z})$. Making use of this approximation, we have:

$$p(\mathbf{y}^*, z^*|\mathbf{x}^*, \Theta, \mathcal{D}) = p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{S}_{z^*}(\mathbf{X}; \widehat{\mathbf{z}}), \mathcal{S}_{z^*}(\mathbf{Y}; \widehat{\mathbf{z}}), \theta_{z^*})p(z^*|\mathbf{x}^*).$$

Finally, we may integrate out z^* to obtain:

$$\begin{aligned} p(\mathbf{y}^*|\mathbf{x}^*, \Theta, \mathcal{D}) &\approx \sum_{z^*} p(\mathbf{y}^*, z^*|\mathbf{x}^*, \Theta, \mathcal{D}) \\ &= \sum_{z^*} p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{S}_{z^*}(\mathbf{X}; \widehat{\mathbf{z}}), \mathcal{S}_{z^*}(\mathbf{Y}; \widehat{\mathbf{z}}), \theta_{z^*})p(z^*|\mathbf{x}^*). \end{aligned}$$

From Eq. (4.23), the predictive distribution $p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{S}_m(\mathbf{X}; \widehat{\mathbf{z}}), \mathcal{S}_m(\mathbf{Y}; \widehat{\mathbf{z}}), \theta_m)$ follows a q-dimensional student-$\mathcal{T}$ distribution:

$$\boldsymbol{p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{S}_m(\mathbf{X}; \widehat{\mathbf{z}}), \mathcal{S}_m(\mathbf{Y}; \widehat{\mathbf{z}}), \theta_m) = \mathcal{T}_q\left(\mathbf{y}^* \left| \mathbf{M}^{(m)}(\mathbf{x}^*, \theta_m), \mathbf{C}^{(m)}(\mathbf{x}^*, \theta_m); n_m - p\right.\right),} \tag{4.94}$$

with the mean and covariance functions given:

$$\begin{aligned}\mathbf{M}^{(m)}(\mathbf{x}^*, \theta_m) &= \mathbf{h}(\mathbf{x}^*)\widehat{\mathbf{B}}_m + \mathbf{a}_m(\mathbf{x}^*)^T \mathbf{A}_m^{-1}(\mathbf{Y}_m - \mathbf{H}_m\widehat{\mathbf{B}}_m), \\ \mathbf{C}^{(m)}(\mathbf{x}^*, \theta_m) &= c^{**}(\mathbf{x}^*, \mathbf{x}^*; \theta_m)\widehat{\Sigma}_m,\end{aligned}$$

with $\widehat{\mathbf{B}}_m$, $\widehat{\Sigma}_m$, and $c^{**}(\cdot)$ defined in Eq. (4.23).

Note the mean function $\mathbf{M}^{(m)}(\mathbf{x}^*, \theta_m)$ is taken as the sampled surrogate model that will be further used to calculate the statistics of interest. $\mathbf{C}^{(m)}(\mathbf{x}^*, \theta_m)$ is the sampled covariance function, which essentially does not contribute to the calculation of the statistics.

A graphical illustration of calculating $p(\mathbf{y}^*|\mathbf{x}^*, \Theta, \mathcal{D})$ is given in Fig. 4.2.

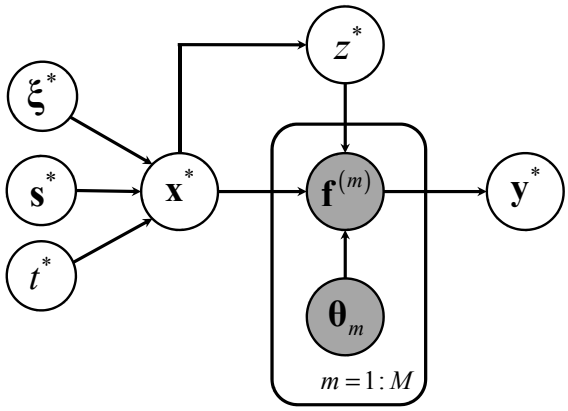

Figure 4.2: Graphical model illustration of doing predictions using the proposed framework. $\mathbf{f}^{(m)}$ and θ_m are known to us from the variational inference algorithm discussed above. $\mathbf{B}_m, \Sigma_m, \mathbf{m}_m, \mathbf{R}_m$ and ν_m are integrated out from the framework. $\mathbf{x}^*$ denotes the new input, z^* gives the predictive responsibilities for each mixture components, and then $\mathbf{y}^*$ is calculated as a weighted combination of the predictions given by each mixture component, as in Eq. (4.94).

Analytic first- and second-order statistics

For one sample Θ', the first and second-order statistics of interest can then be evaluated analytically using the sampled surrogate model $\mathbf{M}^{(m)}(\mathbf{x}^*, \theta'_m)$. From now on, we ignore the dependence on θ'_m in order to lessen the notational burden. The mean at the spatial locations $\mathbf{s}^*$ and time t^* is:

$$\mathbf{M}^*(\mathbf{s}^*, t^*) := \sum_{m=1}^{M} \int \mathbf{M}^{(m)}(\mathbf{x}^*) p(z^* = m|\mathbf{x}^*) p(\xi^*) d\xi^*. \tag{4.95}$$

Recall that $\mathbf{x}^* = (\xi^*, \mathbf{s}^*, t^*)$ and the integration is over only the stochastic inputs ξ^*. This can also be written as:

$$\mathbf{M}^*(\mathbf{s}^*, t^*) = \sum_{m=1}^{M} \left[\zeta_h^{(m)^T} \widehat{\mathbf{B}}_m + \zeta_a^{(m)^T} \mathbf{A}_m^{-1} (\mathbf{Y}_m - \mathbf{H}_m \widehat{\mathbf{B}}_m) \right], \tag{4.96}$$

where

$$\begin{aligned} \zeta_h^{(m)} &= \int \mathbf{h}(\mathbf{x}^*) p(z^* = m|\mathbf{x}^*) p(\xi^*) d\xi^*, \\ \zeta_a^{(m)} &= \int \mathbf{a}_m(\mathbf{x}^*) p(z^* = m|\mathbf{x}^*) p(\xi^*) d\xi^*. \end{aligned}$$

Now let $i, j \in \{1, \ldots, q\}$ be two arbitrary outputs. The covariance matrix between all spatial and time test points is defined by:

$$\begin{aligned} \mathbf{C}_{ij}^*(\mathbf{s}^*, t^*) &:= \int (\mathbf{Y}_i^* - \mathbf{M}_i^*)(\mathbf{Y}_j^* - \mathbf{M}_j^*)^T p(\xi^*) d\xi^*, \\ &= \int \left(\sum_{m=1}^{M} \mathbf{M}_i^{(m)}(\mathbf{x}^*) p(z^* = m|\mathbf{x}^*) - \mathbf{M}_i^* \right) \left(\sum_{r=1}^{M} \mathbf{M}_j^{(r)}(\mathbf{x}^*) p(z^* = r|\mathbf{x}^*) - \mathbf{M}_j^* \right)^T p(\xi^*) d\xi^*, \\ &= \int \left(\sum_{m=1}^{M} \sum_{r=1}^{M} \beta_m \beta_r \mathbf{M}_i^{(m)}(\mathbf{x}^*) \mathbf{M}_j^{(m)^T}(\mathbf{x}^*) \right) p(\xi^*) d\xi^* - \mathbf{M}_i^* (\mathbf{M}_j^*)^T, \end{aligned} \tag{4.97}$$

where $\beta_m = p(\mathbf{z}^* = m|\mathbf{x}^*)$ and $\beta_r = p(\mathbf{z}^* = r|\mathbf{x}^*)$, and the subscripts i and j indicate columns of the associated matrices. This covariance matrix can be evaluated by:

$$\mathbf{C}_{ij}^*(\mathbf{s}^*, t^*) = \sum_{m=1}^{M} \sum_{r=1}^{M} [\kappa_{hh}^{mr} + \kappa_{ha}^{mr} + \kappa_{ah}^{mr} + \kappa_{aa}^{mr}] - \mathbf{M}_i^* (\mathbf{M}_j^*)^T, \tag{4.98}$$

where

$$\begin{aligned}
\kappa_{hh}^{mr} &= \int \beta_m \beta_r \left(\mathbf{h}(\mathbf{x}^*)\widehat{\mathbf{B}}_m\right)_i \left(\widehat{\mathbf{B}}_r^T \mathbf{h}(\mathbf{x}^*)^T\right)_j p(\xi^*) d\xi^*, \\
\kappa_{ha}^{mr} &= \int \beta_m \beta_r \left(\mathbf{h}(\mathbf{x}^*)\widehat{\mathbf{B}}_m\right)_i \left(\widetilde{\mathbf{Y}}_r^T \mathbf{A}_r^{-1} \mathbf{a}_r(\mathbf{x}^*)\right)_j p(\xi^*) d\xi^*, \\
\kappa_{aa}^{mr} &= \int \beta_m \beta_r \left(\mathbf{a}_m(\mathbf{x}^*)^T \mathbf{A}_m^{-1} \widetilde{\mathbf{Y}}_m\right)_i \left(\widetilde{\mathbf{Y}}_r^T \mathbf{A}_r^{-1} \mathbf{a}_r(\mathbf{x}^*)\right)_j p(\xi^*) d\xi^*,
\end{aligned}$$

where $\widetilde{\mathbf{Y}}_m = \mathbf{Y}_m - \mathbf{H}_m \widehat{\mathbf{B}}_m$ and $\kappa_{ah}^{mr} = (\kappa_{ha}^{mr})^T$.

Higher-order statistics can be obtained via Monte Carlo method. By gathering all the sampled response surfaces, one can obtain a probabilistic measure of any statistics of interest with not only mean predictions but also additional error bars. The whole algorithm is summarized in Algorithm 7.

Algorithm 7: Sampling the posterior of statistics of interest.

Require: Observed data $\mathcal{D}$, and approximated posterior $q(\Theta)$,

Ensure: Repeatedly sample from the posterior of statistics

Sample Θ', from the approximated posterior distribution

Sample a response surface and calculate the first and second-order statistics

Interrogate the obtained response surface (analytically or via MC)

4.2 Numerical Examples

In this section, we first study the benchmark Kraichnan-Orszag (KO) problem to demonstrate the unique features of the proposed framework. Then, we continue with the study of a heterogeneous oil reservoir problem. All examples considered are run on massively parallel computers at the National Energy Scientific

Computing Center (NERSCC) [37]. The tolerance δ in Algorithm 5 is taken as $\delta = 0.0001$. The additional tolerances in Algorithm 6 are taken as $\varsigma_1 = 10^{-3}$, and $\varsigma_2 = 10^{-3}$. In the numerical examples considered, the variational inference algorithm converges in no more than 1000 iterations, whereas for similar examples using MCMC [11] more than $100,000$ samples were required to explore the posterior distribution of the parameters.

4.2.1 Kraichnan-Orszag problem

The transformed Kraichnan-Orszag three-mode problem is expressed as the following dynamical system [64, 109]

$$\begin{aligned} \frac{dy_1}{dt} &= y_1 y_3, \\ \frac{dy_2}{dt} &= -y_2 y_3, \\ \frac{dy_3}{dt} &= -y_1^2 + y_2^2, \end{aligned} \tag{4.99}$$

subject to random initial conditions at $t = 0$. A discontinuity in the solution occurs when the initial conditions $y_1(0)$ and $y_2(0)$ cross the plane $y_1 = 0$ and $y_2 = 0$. The deterministic solver we use is a 4-th order Runge-Kutta method as implemented in the GNU Scientific Library [37].

One-dimensional case

Let us first consider the one-dimensional case with initial conditions as

$$\begin{aligned} y_1(0) &= 1, \\ y_2(0) &= 0.1\xi, \\ y_3(0) &= 0, \end{aligned}$$

where $\xi \sim \mathcal{U}([-1, 1])$. This problem has a discontinuity at $\xi = 0$.

The stochastic input ξ, plus the time variable t, determine the input dimension of our model as $d = d_\xi + d_t = 2$. The responses are recorded at 40 equidistant time steps in between the time interval $[0, 10]$, i.e., $n_t = 40$. The hyperparameters are selected as $\alpha = 1$, $\gamma_r = 5$, $\gamma_\epsilon = 10^{-6}$, $\mathbf{u}_0 = \mathbf{0}_d$, $\mathbf{R}_0 = 10^{-3}\mathbb{I}_d$, $\mathbf{W}_0 = 10^{-3}\mathbb{I}_d$ and $\upsilon_0 = d$. The truncation level M is set to 200. The model is trained with $n_\xi = 51$ and $n_\xi = 98$ observations. The observations are initially randomly classified. The variational inference algorithm gradually obtains the optimal clustering by maximizing the lower bound of the evidence. Figure 4.3(a) shows the initial random clustering of the input data. Figure 4.3(b) shows the computed clustering at an intermediate stage of the algorithm. It can be seen that this clustering is not optimized because there are clusters formed with data across the discontinuity ($\xi = 0$). Recall that these clusters not only decompose the stochastic space but also the temporal space. The final clustering pattern obtained is shown in Fig. 4.3(c). Eventually twelve groups are left after running the variational inference algorithm, six on the left side of the discontinuity, and another six on the right side. Note that there is no decomposition of the time space after the algorithm converges. The convergence of the variational inference algorithm can also be seen by plotting the number of clusters at each iteration step (Fig. 4.4). From this figure, we can see that increasing the size of the training data set increases the number of the mixture components. Finally, note that the number of clusters decreases with iteration with some minor oscillation that is possibly due to the local adjustments of the constructed model.

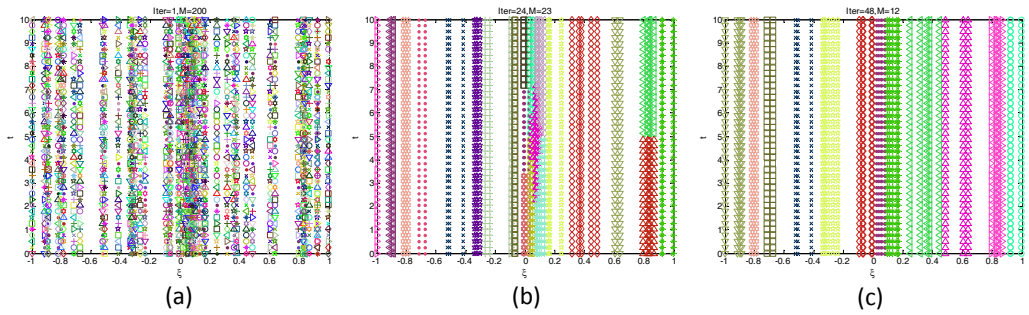

Figure 4.3: KO1 - demonstration of the evolution of the clustering with $n_\xi = 51$ observations: (a) the initial clustering; (b) the clustering at an intermediate iteration of the variational inference algorithm; (c) the final clustering. The number of iterations and the number of clusters selected are shown above each figure.

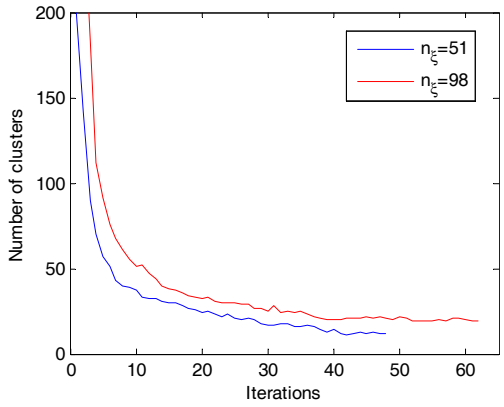

Figure 4.4: KO1 - convergence of the number of clusters with respect to the iteration step for $n_\xi = 51$ and $n_\xi = 98$ observations.

To show the convergence of the posterior distribution, $q(\mathbf{z})$ (Eq. (4.79)), we randomly pick two observations at input points, $(-0.01, 0.8)$ and $(0.9, 6)$, and plot the corresponding normalized responsibilities with respect to all the mixture components (in this example, 200) at each iteration step, as in Fig. 4.5. At the early iterations and for the point $(-0.01, 0.8)$ very close to the discontinuity, note that a number of clusters are taking responsibility in explaining it but eventually only one dominant cluster is selected. For the point $(0.9, 6)$ close to the boundary,

one dominant cluster is selected from the very beginning of the algorithm. This implicitly shows that the approximation of the normalization term $\widehat{C}_n(\widehat{\mathbf{m}}, \widehat{\mathbf{R}}, \widehat{\nu})$ in Eq. (4.79) does not affect the convergence of the variational inference algorithm. The convergence of the lower bound of the model evidence is given in Fig. 4.6.

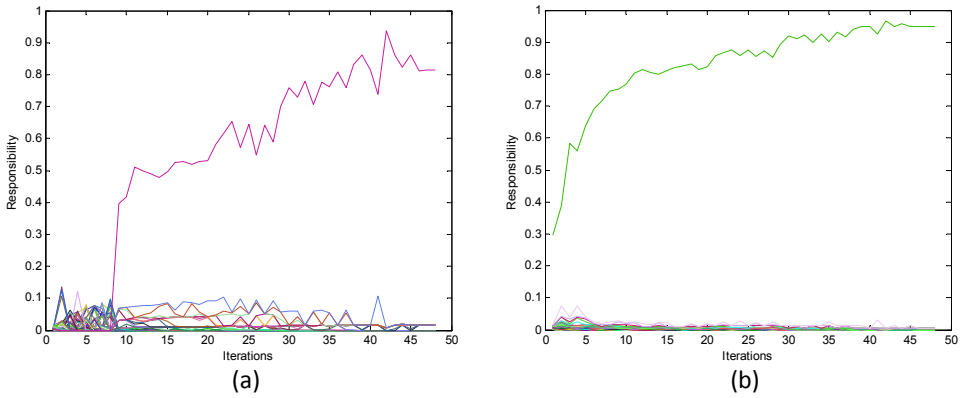

Figure 4.5: KO1 - convergence of the responsibility for $n_\xi = 51$ observations: (a) for input point $(-0.01, 0.8)$; (b) for input point $(0.9, 6)$. We start with $M = 200$ components, and as the variational inference algorithm proceeds most of the components vanish and therefore do not contribute to the responsibility of the corresponding component for a query point. The algorithm eventually provides one dominant component for each query point.

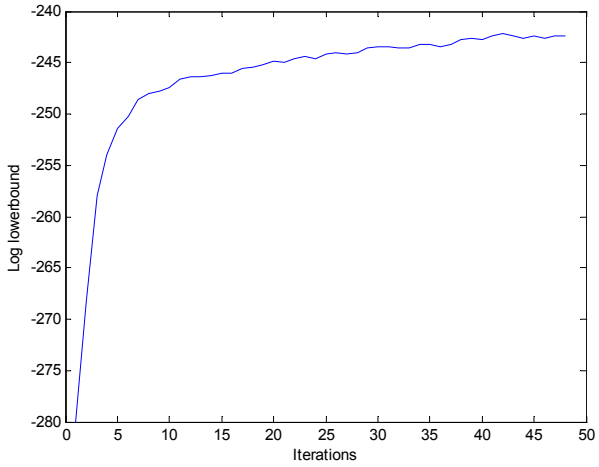

Figure 4.6: KO1 - convergence of the lower bound for the model evidence.

The variational inference algorithm converges fast because the update equations for the proposed posterior distributions $q(\nu)$ (Eq. (4.63)) and $q(z)$ (Eq. (4.79)) are analytical. The update of $q(\theta)$ is a bit more complex but it does not deteriorate the performance of the variational inference algorithm due to the nature of the gradient descent algorithm.

From the variational inference algorithm, the approximated posterior distribution of parameters of interest can be obtained. Then we draw 100 samples from the approximated posterior distribution and we make predictions of the statistics of interest. For each parameter sample, the predictions are made at 100 equidistant time steps ($n_t^* = 100$) in $[0, 10]$. The statistics of interest can be calculated semi-analytically, using Eqs. (4.96) and (4.98). We then calculate the mean of the statistics as well as the 95% confidence intervals (error bars). With $n_\xi = 98$ observations, the constructed model can capture the mean and also the variance of each output quite accurately, as shown in Fig. 4.7.

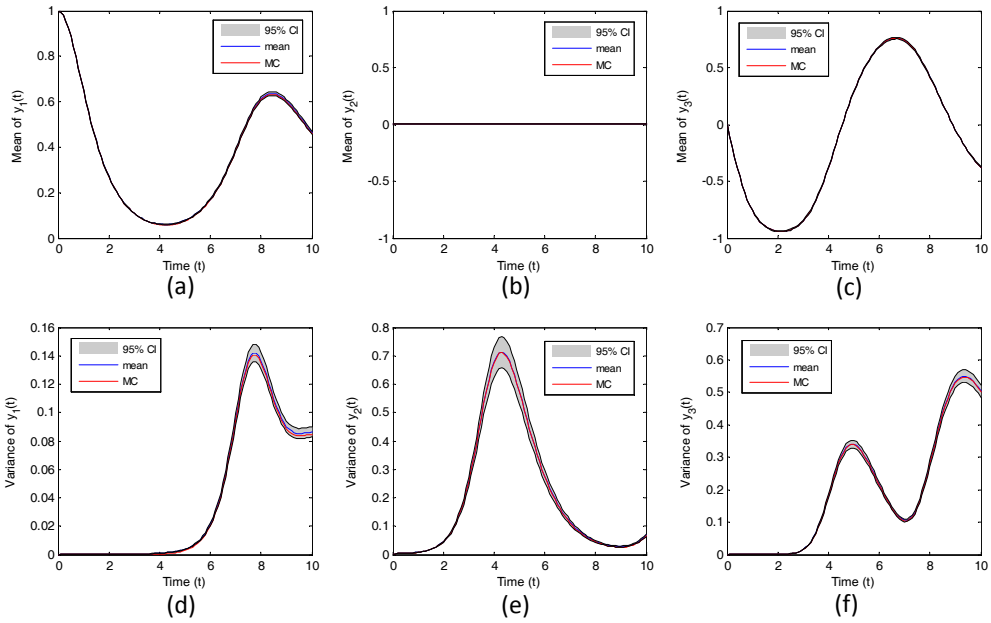

Figure 4.7: KO1 - The blue curve is the mean of the statistic of interest (mean and variance) predicted by the model while the gray area shows 95% confidence intervals. The red curve is the MC result using 10^6 samples. The first row provides the predictive means for y_1, y_2 and y_3 with $n_\xi = 98$. The second row shows the corresponding predictive variances.

We further evaluate the probability density functions (PDFs) of the responses at certain time steps. The predictive PDFs are obtained in a similar way as discussed above: (1) we sample 100 parameters from the approximated posteriors; (2) for each set of the parameters, we draw $50,000$ samples from $p(\xi)$; (3) we predict the response for each of these ξ; and (4) we use kernel density estimator to approximate the desired PDF [75]. The mean of the predicted PDFs with $n_\xi = 98$ for each output at various time locations as well as the error bars are compared to the MC results obtained with 10^6 samples as shown in Fig. 4.8.

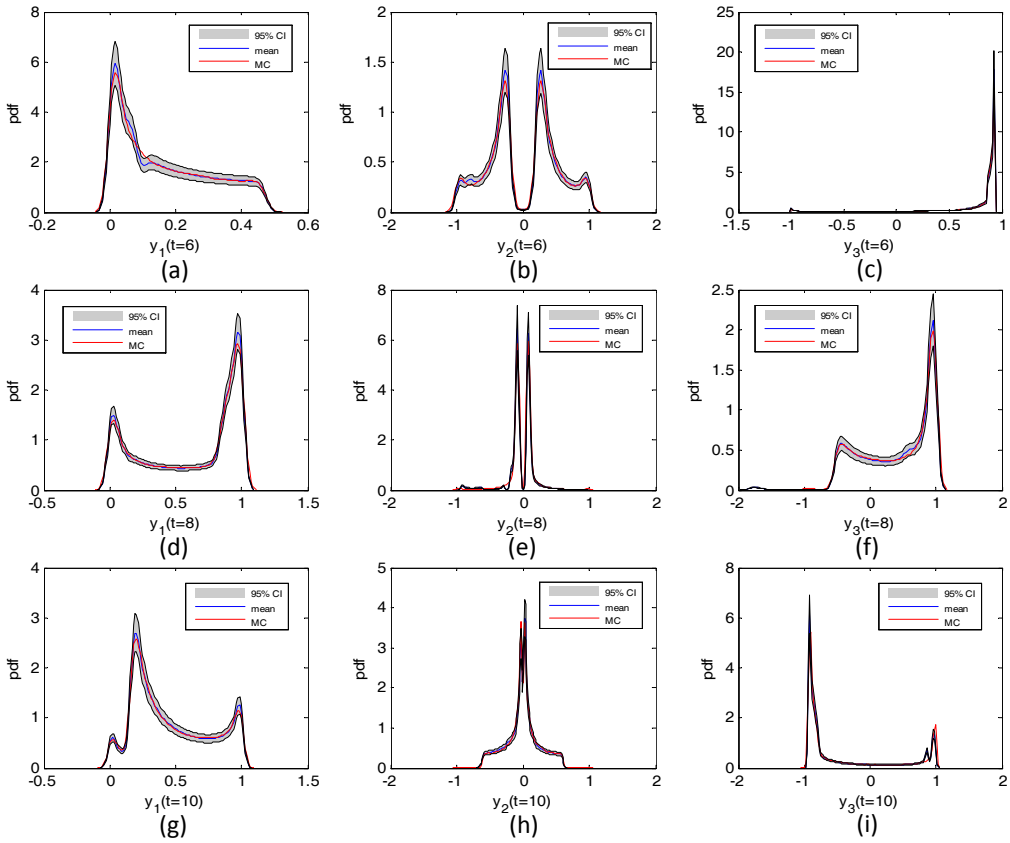

Figure 4.8: KO1 - Comparison of the predicted PDFs with $n_\xi = 98$ to the PDFs obtained with MC. Each row depicts the PDFs of y_1, y_2 and y_3 for time $t = 6, 8, 10$, respectively. The blue curve is the mean of the statistic predicted by the model while the gray area shows 95% confidence intervals. The red curve is the MC estimate obtained using 10^6 samples.

In comparison with the multi-element GP approach in [9], the present method can automatically cluster the observations, and the discontinuities can be discovered without explicitly decomposing the stochastic space. Due to the nature of the variational inference algorithm, the posterior distributions of interest are more efficiently approximated than using MCMC [11]. Also by the nature of the mixture model, less data are assigned to each GP component thus avoiding forming large size covariance matrices.

Two-dimensional case

Let us now consider the two-dimensional case. The initial conditions for the problem are taken as:

$$\begin{aligned} y_1(0) &= 0, \\ y_2(0) &= 0.1\xi_1, \\ y_3(0) &= \xi_2, \end{aligned}$$

where

$$\xi_i \sim \mathcal{U}([-1,1]),\ i = 1,2.$$

The input dimension for the 2D problem is $d = d_\xi + d_t = 3$. The responses are also recorded at 40 equidistant time steps in between the time interval $[0, 10]$, i.e., $n_t = 40$. The hyper-parameters are selected as for the 1D case. The truncation level M is set to be 800. The model is trained with $n_\xi = 100$, 200 and 400 observations and the algorithm stops at $52, 85$, and 112 clusters, respectively. The decomposition of the time space has been observed around the discontinuity when the algorithm converges. Figure 4.9 shows the comparison of the mean predictions of the mean of each output as well as the error bars with $n_\xi = 200$ to the MC estimates with 10^6 samples. As shown from the figure, the mean of each output is very accurately captured using the constructed model with $n_\xi = 200$. The variance of each output is gradually captured with an increase of the number of observations, and the corresponding error bars are also shrinking, as shown in Fig. 4.10. The predicted probability density functions (PDFs) for the responses are obtained in a similar way to that described in the 1D example. Here, we only compare the predicted PDFs for y_2 at various time locations

to the MC results. The mean predictions as well as the error bars for each output at various times are compared to the MC estimates, as shown in Fig. 4.11. From these figures, even with few numbers of observations, the discontinuity can be discovered, and as expected increasing the number of observations results in more accurate predictions.

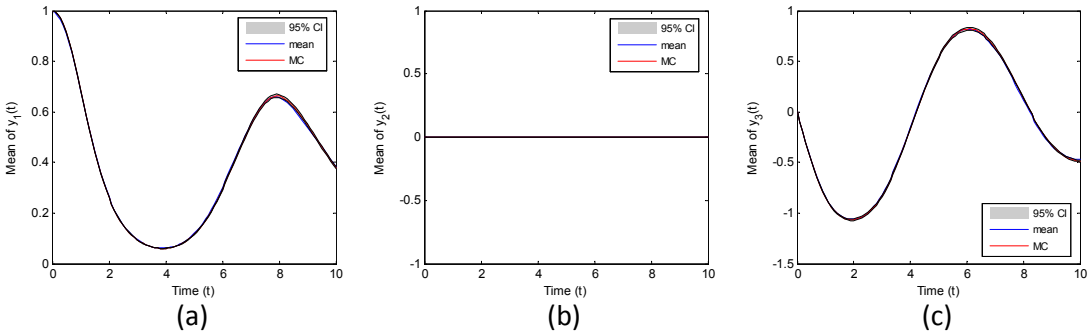

Figure 4.9: KO2 - Comparison of predictive means for each output with the MC computed means for $n_\xi = 200$. The blue curve is the mean of the statistic predicted by the model while the gray area shows 95% confidence intervals. The red curve is the MC estimate obtained using 10^6 samples.

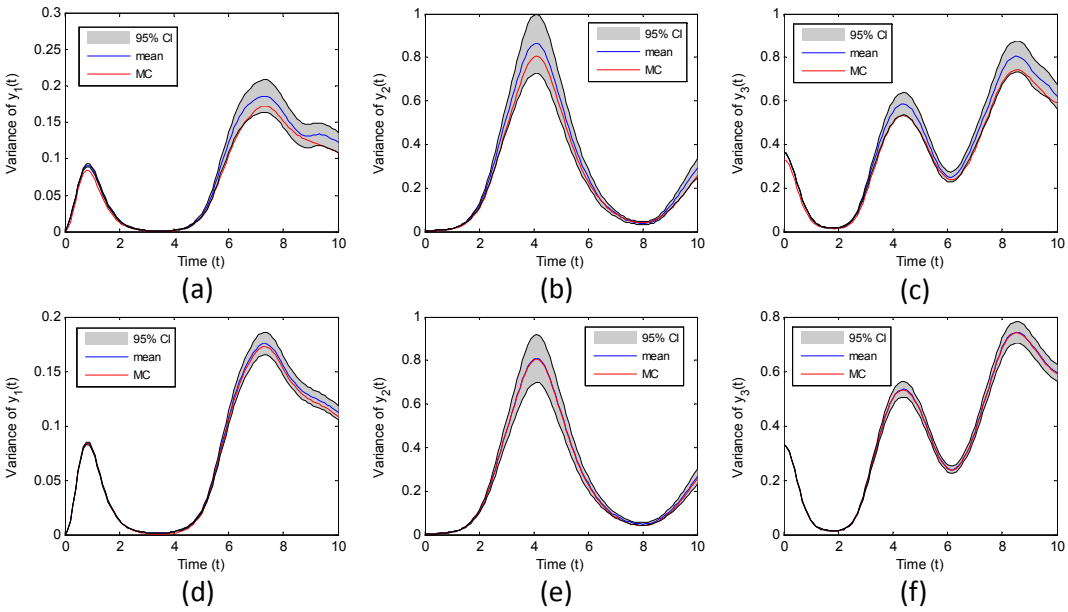

Figure 4.10: KO2 - Comparison of predictive variances for each output with the MC computed means. The top row provides the results with $n_\xi = 200$, and the bottom row gives the predictions with $n_\xi = 400$. The blue curve is the mean of the statistic predicted by the model while the gray area shows 95% confidence intervals. The red curve is the MC result obtained using 10^6 samples.

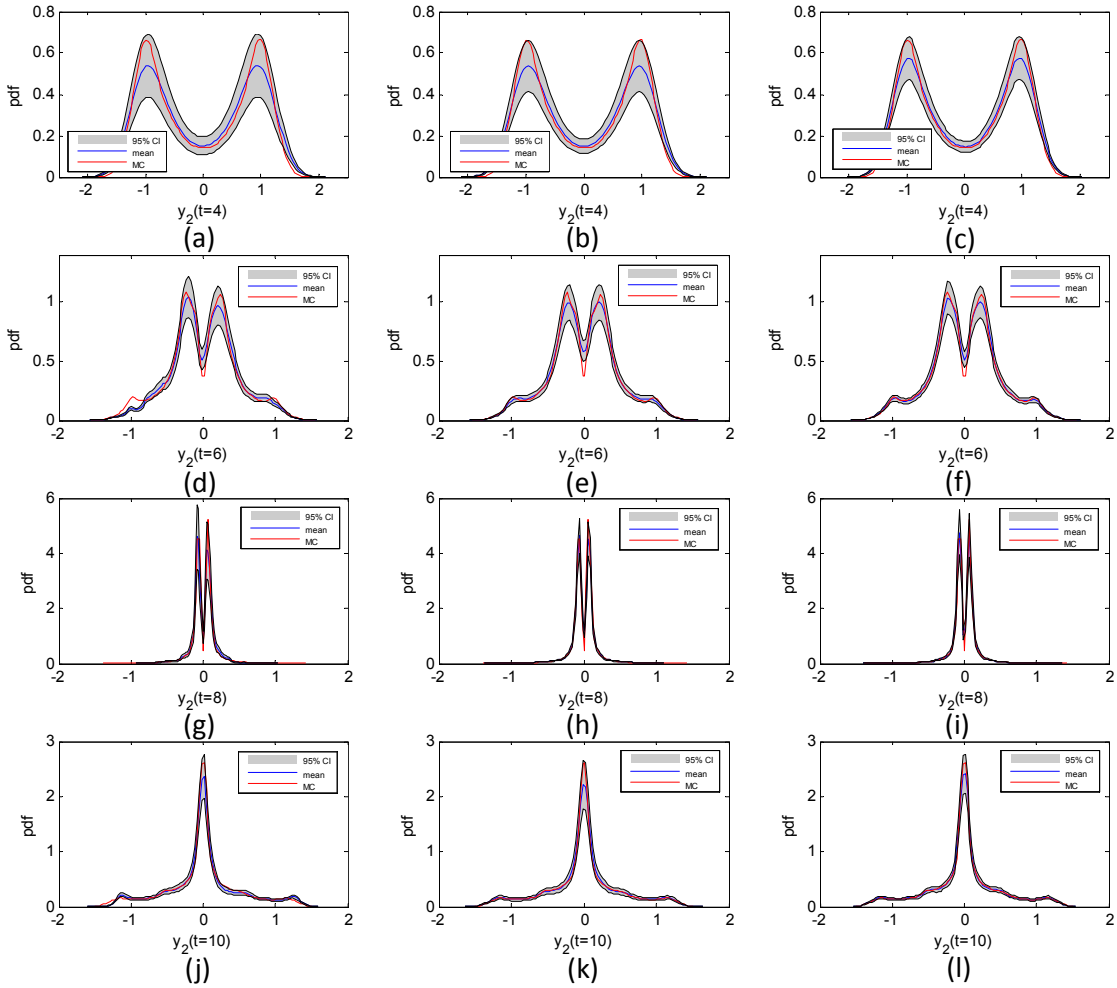

Figure 4.11: KO2 - Convergence plots of PDFs for y_2 at different time steps and different numbers of observations. The first column corresponds to $n_\xi = 100$, the second to $n_\xi = 200$, and the third to $n_\xi = 400$. Each row depicts the PDF of $y_2(t)$ at times $t = 4, 6, 8, 10$. The blue curve is the mean of the statistic predicted by the model while the gray area shows 95% confidence intervals. The red one is the MC estimates with 10^6 samples.

Three-dimensional case

Finally, let us consider the more difficult and computationally demanding three-dimensional case. The initial conditions for the problem are:

$$y_1(0) = \xi_1,$$
$$y_2(0) = \xi_2,$$
$$y_3(0) = \xi_3,$$

where

$$\xi_i \sim \mathcal{U}([-1, 1]),\ i = 1, 2, 3.$$

The hyperparameters are set as in the previous cases, the truncated level M is set to 2000. The model is trained with $n_\xi = 1000$, and $n_\xi = 4000$ observations. For this case, the algorithm stops at 376, and 982 clusters for $n_\xi = 1000$ and $n_\xi = 4000$, respectively. The decomposition of the time space has also been observed at convergence. Again, the constructed models can capture the mean and variance of the statistics of interest very well. As shown in Fig. 4.12, the increase of the number of observations results in a better predictive performance of the constructed model. Similarly, we compare the predictive PDFs for y_2 and y_3 at $t = 8, 10$ with $n_\xi = 4000$ to the MC results with 10^6 samples. From Fig. 4.13, we can see that the predicted PDFs agree well with the MC estimates as expected.

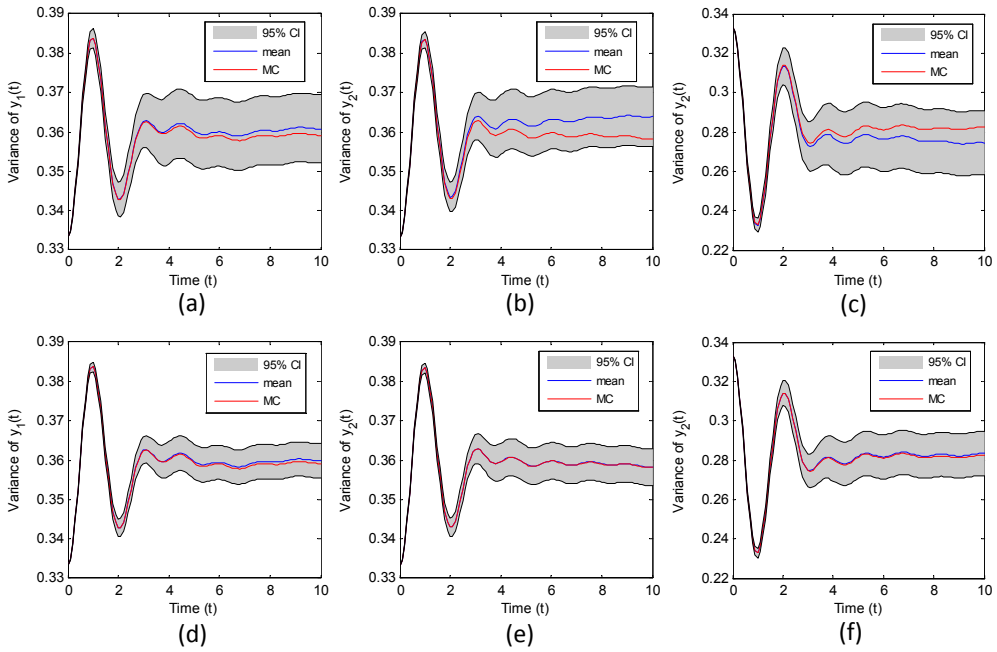

Figure 4.12: KO3 - Comparison of predictive variances for $y_1(t)$, $y_2(t)$, and $y_3(t)$ with $n_\xi = 1000$ and 4000 to the MC variances with $n_\xi = 10^6$. The blue curve is the mean of the predicted variance while the gray area shows 95% confidence intervals. The red curve is the MC result.

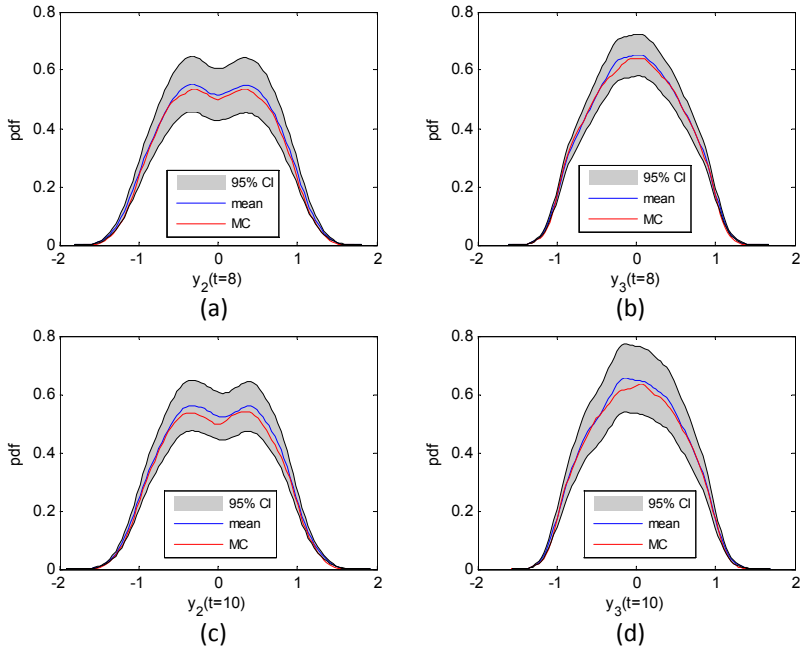

Figure 4.13: KO3 - Comparison of predictive PDFs for y_2 and y_3 at $t = 8, 10$ with $n_\xi = 4000$. The first row corresponds to $t = 8$, the second to $t = 10$. The blue curve is the mean of the statistic predicted while the gray area shows 95% confidence intervals. The red curve is the MC estimate with 10^6 samples.

4.2.2 Flow through porous media

In this section, we study a heterogeneous oil reservoir flow problem in a confined domain Λ in $\mathbb{R}^{d_s}$. A standard quarter-five spot problem with one injection well on the bottom left and one production well on the top right is considered. The flow is taken as an immiscible two-phase system with water and oil that is incompressible [40, 3, 2]. The capillary pressure and gravity are not included in the model.

From Darcy's law, we can first write the *pressure equation* for the flow as:

$$\nabla \cdot \boldsymbol{v} = q, \text{ where } \boldsymbol{v} = -(\lambda_o + \lambda_w)\mathbf{K}\nabla p, \tag{4.100}$$

where the subscripts o, w represent oil phase and water phase, respectively. $\boldsymbol{v}$ is the total velocity, i.e., $\boldsymbol{v} = \boldsymbol{v}_o + \boldsymbol{v}_w$. $q = q_w + q_o$ is the source/sink term, which models the injection/production well in this problem. No-flow boundary conditions are considered. $\mathbf{K}$ is the location-dependent permeability tensor. p is the pressure, and here we take $p_o = p_w = p$ due to the assumption of no capillary pressure. λ_α, with $\alpha = o, w$, is the phase mobility given by $\lambda_\alpha = k_{r\alpha}/\mu_\alpha$, where $k_{r\alpha}(s)$ is the relative permeability depending on the saturation s_α (fraction of the void occupied by phase α), and μ_α is the viscosity of phase α. The relative permeability models the reduced conductivity of a phase due to the presence of other phases, and is, according to common practice, assumed to be a function of the saturation only [3]. In this work, we use

$$k_{rw} = (s')^2, \quad k_{ro} = (1 - s')^2, \quad s' = \frac{s - s_{wc}}{1 - s_{wc} - s_{or}}, \tag{4.101}$$

with $s_{wc} = s_{or} = 0.2$ and initial saturation $s_0 = s_{wc}$. Also, s_{or} is the irreducible oil saturation, i.e., the lowest oil saturation that can be achieved by displacing oil by water, and s_{wc} is the connate water saturation, i.e., the saturation of water trapped in the pores of the rock during formation of the rock.

By introducing the concept of *fractional flow* $f_\alpha = \lambda_\alpha/\lambda$, where $\lambda = \lambda_o + \lambda_w$ is the total mobility, we can write the phase velocity $\boldsymbol{v}_\alpha = f_\alpha \boldsymbol{v}$. Note f_α is also dependent on the saturation s_α. Then, combined with the conservation equation of mass for each phase α, we may write the *saturation equation* (fluid transport), as follows:

$$\phi \frac{\partial s_\alpha}{\partial t} + \nabla \cdot (f_\alpha \boldsymbol{v}) = q_\alpha, \tag{4.102}$$

where ϕ is porosity, which is modeled as a random field in this problem rather than a constant as in [66, 11]. The porosity has been shown to have a strong correlation with the permeability, as discussed in Chapter 1. Since $s_o + s_w = 1$,

only the water saturation s_w is considered below.

The fraction of water/oil in the produced fluid as a function of time, $F(t)$, is of great interest [66, 40], and it known as the *water-cut curve* defined as

$$F(t) = \frac{\int_{\partial\Lambda_{out}} f_w(\boldsymbol{v}\cdot\boldsymbol{n})d\boldsymbol{x}}{\int_{\partial\Lambda_{out}} (\boldsymbol{v}\cdot\boldsymbol{n})d\boldsymbol{x}}, \tag{4.103}$$

where $\partial\Lambda_{out}$ is the outflow boundary, $\boldsymbol{n}$ is the normal to the boundary. Also note that here t is measured in days. This is different from the traditional learning of the water-cut curve [66], where t is considered as a dimensionless time (PVI, pore volume injected). PVI is not considered in this paper since the porosity is also modeled as a random field (thus, for different realizations, if we consider the same amount of water injected, the corresponding PVI will be different).

Discretizing the pressure equation

The pressure equation is discretized with a mixed FEM method [24, 3]. Recall in this work, we consider no flow boundary condition, i.e., $\boldsymbol{v}\cdot\boldsymbol{n} = 0$ and an extra constraint, $\int_\Lambda p d\boldsymbol{x} = 0$, is added to close the system. To derive the mixed formulation, we first define the Sobolev space

$$H_0^{div}(\Lambda) = \left\{\boldsymbol{v} \in \left(L^2(\Lambda)\right)^{d_s} : \nabla\cdot\boldsymbol{v} \in L^2(\Lambda) \text{ and } \boldsymbol{v}\cdot\boldsymbol{n} = 0 \text{ on } \partial\Lambda\right\}. \tag{4.104}$$

The mixed-finite element discretization in domain Λ seeks a pair $(\boldsymbol{v}, p) \in U \times V$, where U and V are finite-dimensional subspaces of $H_0^{div}(\Lambda)$ and $L^2(\Lambda)$, respectively, such that,

$$\int_\Lambda \boldsymbol{v}\cdot(\lambda\mathbf{K})^{-1}\boldsymbol{u}d\boldsymbol{x} - \int_\Lambda p\nabla\cdot\boldsymbol{u}d\boldsymbol{x} = 0, \qquad \text{for all } \boldsymbol{u}\in U, \tag{4.105}$$

$$\int_\Lambda l\nabla\cdot\boldsymbol{v}d\boldsymbol{x} = \int_\Lambda lqd\boldsymbol{x}, \qquad \text{for all } l\in V. \tag{4.106}$$

Now, let us partition the domain Λ into mutually disjoint grid cells as $\Lambda = \{\Lambda_m\}$. The basis functions χ_m and ψ_{ij} on a cell Λ_m and edge $\gamma_{ij} = \partial\Lambda_i \cap \partial\Lambda_j$ are defined respectively by:

$$\chi_m = \{ 1 \ , \text{ if } \boldsymbol{x} \in \Lambda_m, 0, \text{ elsewise,} \tag{4.107}$$

and

$$\boldsymbol{\psi_{ij}} \cdot \boldsymbol{n_{kl}}|_{\gamma_{kl}} = \{ \mathbf{1} \ , \text{ if } \boldsymbol{\gamma_{ij}} = \boldsymbol{\gamma_{kl}}, \mathbf{0}, \text{ elsewise} \ . \tag{4.108}$$

Thus, we can write $p = \sum p_m \chi_m$ and $\boldsymbol{v} = \sum v_{ij} \psi_{ij}$, where $v_{ij} = \int_{\gamma_{ij}} \boldsymbol{v} \cdot \boldsymbol{n}_{ij} d\boldsymbol{x}$, where $\boldsymbol{n}_{ij}$ is the unit normal to γ_{ij} from Λ_i to Λ_j. This allows us to rewrite the pressure equation as a linear system in $\mathbf{p} = \{p_m\}$ and $\mathbf{v} = \{v_{ij}\}$. This system takes the following form:

$$\begin{bmatrix} \mathbf{B} & -\mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{q} \end{bmatrix}. \tag{4.109}$$

Here, the **B** and **C** blocks are defined as follows:

$$b_{ij,kl} = \int_\Lambda \psi_{ij} \cdot (\lambda \mathbf{K})^{-1} \psi_{kl} d\boldsymbol{x}, \tag{4.110}$$

$$c_{m,kl} = \int_\Lambda \chi_m \nabla \cdot \psi_{kl} d\boldsymbol{x}, \tag{4.111}$$

$$q_m = \int_\Lambda \chi_m q d\boldsymbol{x}. \tag{4.112}$$

Discretizing the saturation equation

The saturation is evolved in time with a finite-volume scheme and an implicit time discretization. Consider a cell Λ_i with edges γ_{ij}, then the finite-volume scheme takes the following form:

$$s_i^{n+1} = s_i^n + \frac{\Delta t}{\phi_i |\Lambda_i|} \left(Q_i(s^{n+1}) - \sum_{j \neq i} F_{ij}(s^{n+1}) v_{ij} \right), \tag{4.113}$$

where Δt denotes the time step and $|\Lambda_i|$ is the measure of grid cell Λ_i. ϕ_i is the porosity in Λ_i and is a constant on the cell. $Q_i(s^{n+1}) = \int_{\Lambda_i} q_w(s_w^{n+1}) d\boldsymbol{x}$ is the source contribution in Λ_i, and

$$F_{ij}(s^{n+1}) = \max \left\{ \text{sign}(v_{ij}) f_w(s_i^{n+1}), -\text{sign}(v_{ij}) f_w(s_j^{n+1}) \right\} \tag{4.114}$$

is the upstream-weighted fractional flow function at γ_{ij}, where s_i^{n+1} is the saturation in Λ_i at time step t_{n+1}.

A Newton-Raphson iterative method is employed to solve the implicit system of Eq. (4.113) as in [1]. The initial guess is chosen to be the saturation field from the previous time step.

Following [3], the system of Eqs. (4.100) and (4.102) is solved by a sequential splitting method (IMPES) [31]. The algorithm in brief is as follows: first, the saturation from the previous time step is used to compute the saturation-dependent variables, then the pressure equation of Eq. (4.100) is solved with the mixed FEM method. Next, the velocity is kept as a constant and used to evolve the saturation to the next time step with an upstream-weighted finite-volume method. The whole algorithm is summarized in Algorithm 8.

In the numerical example to follow (Section 4.2.2), we take $\mu_w = 0.3\ cP$, $\mu_o = 3.0\ cP$, $s_{wc} = s_{or} = 0.2$ and initial saturation $s_0 = s_{wc}$. Note, that in general, we set two time steps Δt_p and Δt_s, one for the evolution of $\mathbf{p}$ and $\boldsymbol{v}$, and the other one for updating the saturation s, respectively. The pressure and velocity evolve slower than the saturation, therefore, in practice, we often set $\Delta t_p \gg \Delta t_s$.

Algorithm 8: IMPES for solving the two-phase flow problem

Initialize: Set $t = 0$, s_{wc}, s_{or}, $s_0 = s_{wc}$, μ_w, μ_o

Set Δt_p **and** Δt_s

while $t \leq T$ **do**

  Calculate λ_w^t **and** λ_o^t **using Eq. (4.101)**

  Solving $\mathbf{p}^t$ **and** $\boldsymbol{v}^t$ **from Eq. (4.109)**

  Set $t_1 = 0$

  while $t_1 \leq \Delta t_p$ **do**

    Solving $s_w^{t+t_1}$ **from Eq. (4.113) using** $\boldsymbol{v}^t$

    Set $t_1 = t_1 + \Delta t_s$

  end while

  Set $t = t + \Delta t_p$

end while

Parametrization of uncertainty

In this work, we consider that the input uncertainty comes from both the permeability and porosity of the oil reservoir. In earlier studies [66, 11, 9], only the random permeability was considered. In general, the permeability and porosity are strongly correlated. This is apparent from the SPE-10 data set [25], as shown in Fig. 4.14.

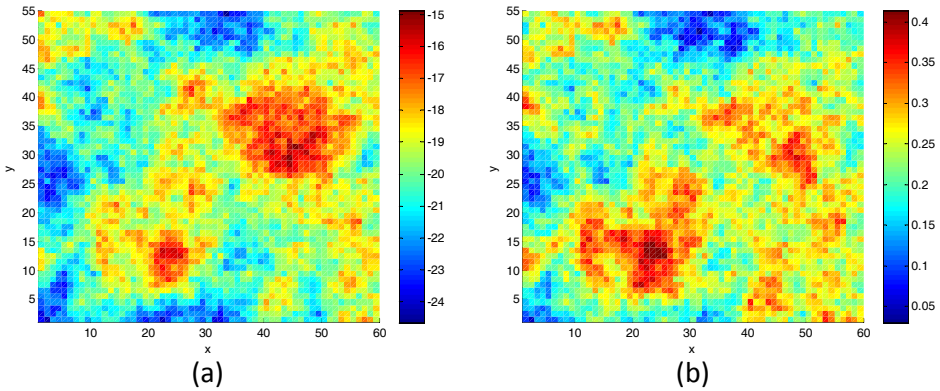

Figure 4.14: Porous media flow - one measurement from the SPE-10 data set, layer one: (a) log-permeability; (b) porosity.

To parameterize the uncertainty, we need to construct a stochastic input model based on available observations. Here, we use the SPE-10 data set [25], which is measured in the range of $1200 \times 2200 \times 170$ (ft^3) and discretized in a regular Cartesian grid with $60 \times 220 \times 85$. In this work, we decompose the observations into 340 reservoir domains with each one in a 60×54 grid. Each observation includes the log-permeability and log-porosity on the 60×54 grid. We assume that these 340 observations follow a second-order random field $G(\boldsymbol{x}, \omega)$. The Karhunen-Loève expansion (KLE) [62] is employed to approximate this random field with a finite-dimensional representation:

$$G(\boldsymbol{x}, \omega) = \mathbb{E}[G(\boldsymbol{x})] + \sum_{i=1}^{N} \sqrt{\lambda_i} \phi_i(\boldsymbol{x}) \xi_i, \tag{4.115}$$

where $\{\xi_i\}_{i=1}^{N}$ are uncorrelated random variables here taken to follow the standard normal $\mathcal{N}(0, 1)$ distribution. $\mathbb{E}[G(\boldsymbol{x})]$ is the mean of the 340 observations. The covariance function is obtained from these observations as well and then a eigenvalue problem is solved to compute the eigenfunctions $\phi_i(\boldsymbol{x})$ and eigenvalues λ_i.

We select the first 100 dimensions to describe the SPE-10 measurements,

which preserves a total of 91% energy from Fig. 4.15. Fig. 4.16 gives an example of simultaneously sampled permeability and porosity generated from the constructed stochastic input model.

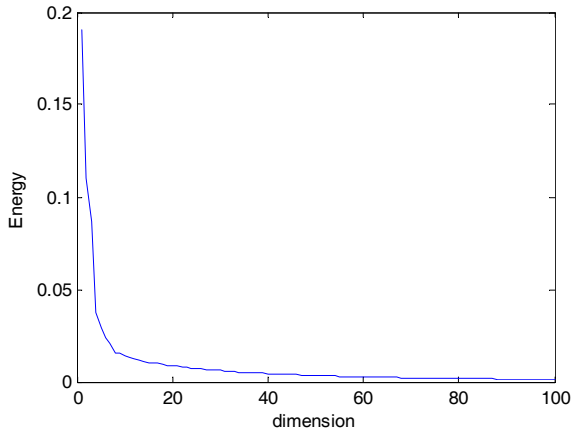

Figure 4.15: Porous media flow - The normalized energy plot for the SPE-10 measurements. Here,"normalized" means that each eigenvalue is divided by the sum of all eigenvalues.

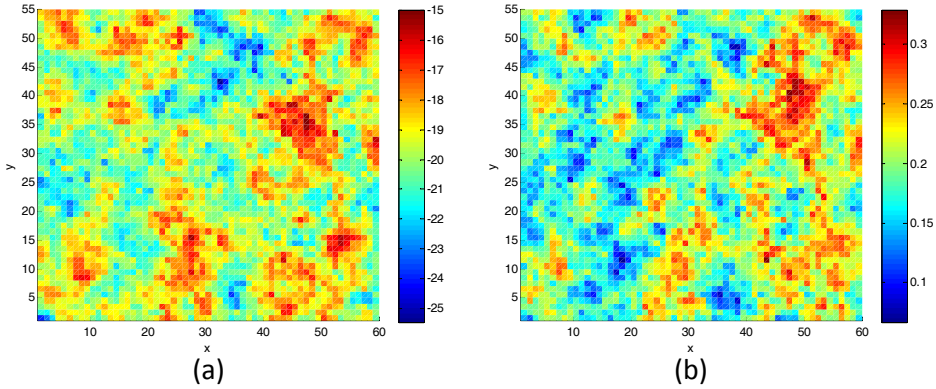

Figure 4.16: Porous media flow - Sampled permeability and porosity field by the constructed stochastic input model: (a) log-permeability; (b) porosity.

Numerical results

The input stochastic dimension is set as 100. For a 2D problem, $d_s = 2$ and the input dimension of the model is $d = d_\xi + d_s + d_t = 103$. The responses under consideration are the velocity components, the pressure and the saturation. As discussed above, our deterministic flow solver is the source driven model, with an injection well at the bottom left and production well at the top right. Therefore, the source term q in the pressure equation (Eq. (4.100)) is defined as a vector with only two non-zero components, one on the bottom left cell and the other one on the top right cell. These two values can be freely set in principle as they will only affect the speed of saturation in the reservoir domain. In the SPE10 projection [25], the water injection rate was set as 5000 fluid barrel per day and the reservoir domain was fully saturated around 2000 days. Therefore, in this paper, we set $q = \pm 2$ for injection/production wells to mimic this process.

The time range considered is $[0, 2100]$ days. For each stochastic input ξ, the response is observed on 30×27 square spatial grid, and recorded every 30 days, i.e., $n_t = 70$. The hyper-parameters are selected as $\alpha = 1$, $\gamma_r = 10$, $\gamma_\epsilon = 100$, $\mathbf{u}_0 = \mathbf{0}_d$, $\mathbf{R}_0 = 10^{-3}\mathbb{I}_d$, $\mathbf{W}_0 = 10^{-3}\mathbb{I}_d$ and $\upsilon_0 = d$. The truncation level M is set to be $20,000$. The model is trained with $n_\xi = 40, 80$ and 160 observations, where the stochastic input ξ is randomly sampled from $\mathcal{N}(0,1)^{n_\xi}$. The approximated posterior distribution of the Dirichlet process converges at thousands of clusters. This agrees with the findings in [97, 98] that with a large number of observations, the DP tends to converge at a state with a larger number of clusters (some of similar sizes). Decomposition over the spatial domain and temporal domain is also observed as expected. Due to the high-dimensionality of the input, it is hard to plot the decomposition of the space. One could only observe certain patterns

in the data clustering. For the spatial region around the source (bottom-left), the algorithm tends to decompose the spatial domain rather than the temporal domain, and for other regions, it is the opposite. No specific patterns of the stochastic space were observed.

Remark 8**.** Several computational improvements can be considered in the way the local covariance matrix is calculated. For example, the separable framework in [11] can be easily integrated with this work to simplify the calculation of the local covariance matrix. This requires certain modifications in the calculation of the posterior distribution of θ_m and in the calculation of statistics.

Remark 9**.** To avoid a memory leak problem, a parallelized structure is needed. Given a certain number of computer nodes, the remaining mixture components at each variational inference step are equally distributed among these nodes. The update of $q(\theta^{(m)})$ for the m-th component is done within its storing node, while the update of $q(v)$, $q(\mathbf{m})$, $q(\mathbf{R})$ and $q(z)$ is done on the root node. The predictions given by each component is done locally in parallel while the assembly of all of these predictions is executed on the root node. This parallelized algorithm requires constant communication between the core node and other nodes, but it accelerates the prediction speed and resolves the memory issue.

After obtaining the approximations to the posterior distributions of interest from the variational inference algorithm, we draw 100 samples from them and construct the corresponding 100 surrogate models. For each sampled surrogate, the predictions are made on a 60×54 grid and every 20 days. The statistics of interest can be calculated semi-analytically, using Eqs. (4.96) and (4.98). The mean and the standard deviation of the statistics of interest could be cal-

culated by using the 100 sampled surrogates. The predictions are compared to the Monte Carlo estimates with 10^5 observations. Fig. 4.17 compares the mean predictions of the mean of the saturation with different sizes of the training data set at $T = 1000$ days to the MC estimate. Two standard deviations (error bars) of the mean of the saturation for the case of $n_\xi = 160$ is provided in Fig. 4.17(d). Fig. 4.18 compares the mean predictions of the variance of the saturation at $T = 1000$ days to the MC estimate. The error bars for $n_\xi = 160$ are also provided. The same statistics are reported for the saturation at $T = 2000$ days in Figs. 4.19 and 4.20. Also, Figs. 4.21-4.24 show the mean predictions of the mean and standard deviation for the velocity components at $T = 2000$ days and compare them to the MC estimates. To allow reasonable visualization of the highly heterogeneous nature of the shown fields, the natural logatirhm of the velocity components is plotted.

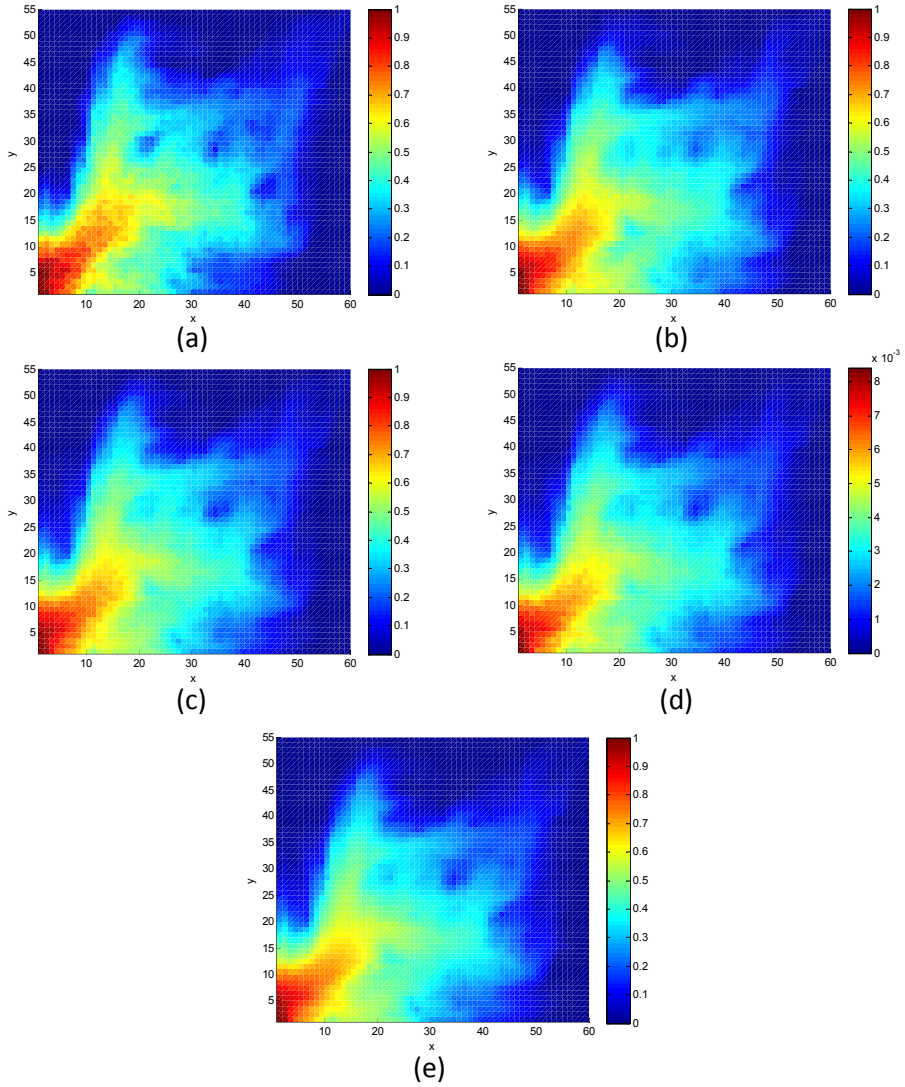

Figure 4.17: Porous media flow - comparison of mean predictions of the mean of the saturation at $T = 1000$ days provided by the model with (a) $n_\xi = 40$; (b) $n_\xi = 80$; (c) $n_\xi = 160$, to the MC result with (e) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the mean of the saturation at $T = 1000$ days for $n_\xi = 160$ observations.

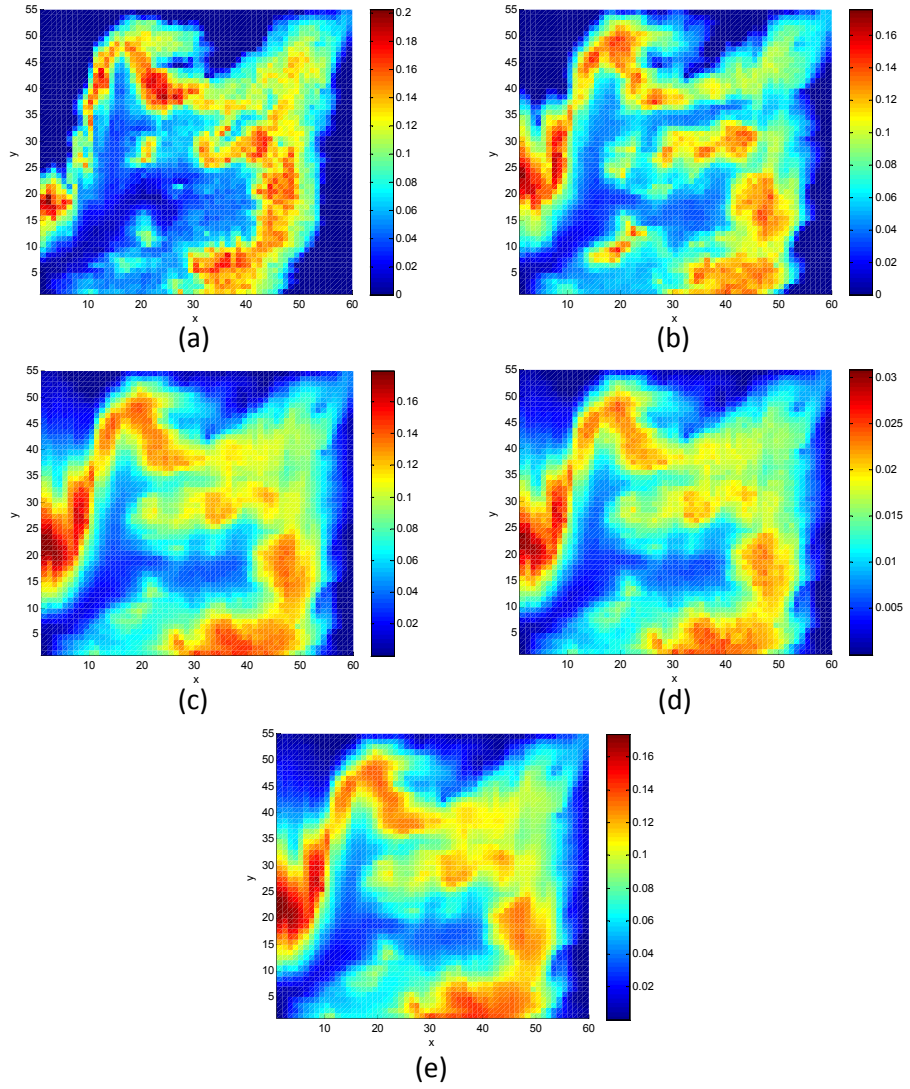

Figure 4.18: Porous media flow - comparison of mean predictions of the std of the saturation at $T = 1000$ days provided by the model with (a) $n_\xi = 40$; (b) $n_\xi = 80$; (c) $n_\xi = 160$, to the MC result with (e) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the std of the saturation at $T = 1000$ days for $n_\xi = 160$ observations.

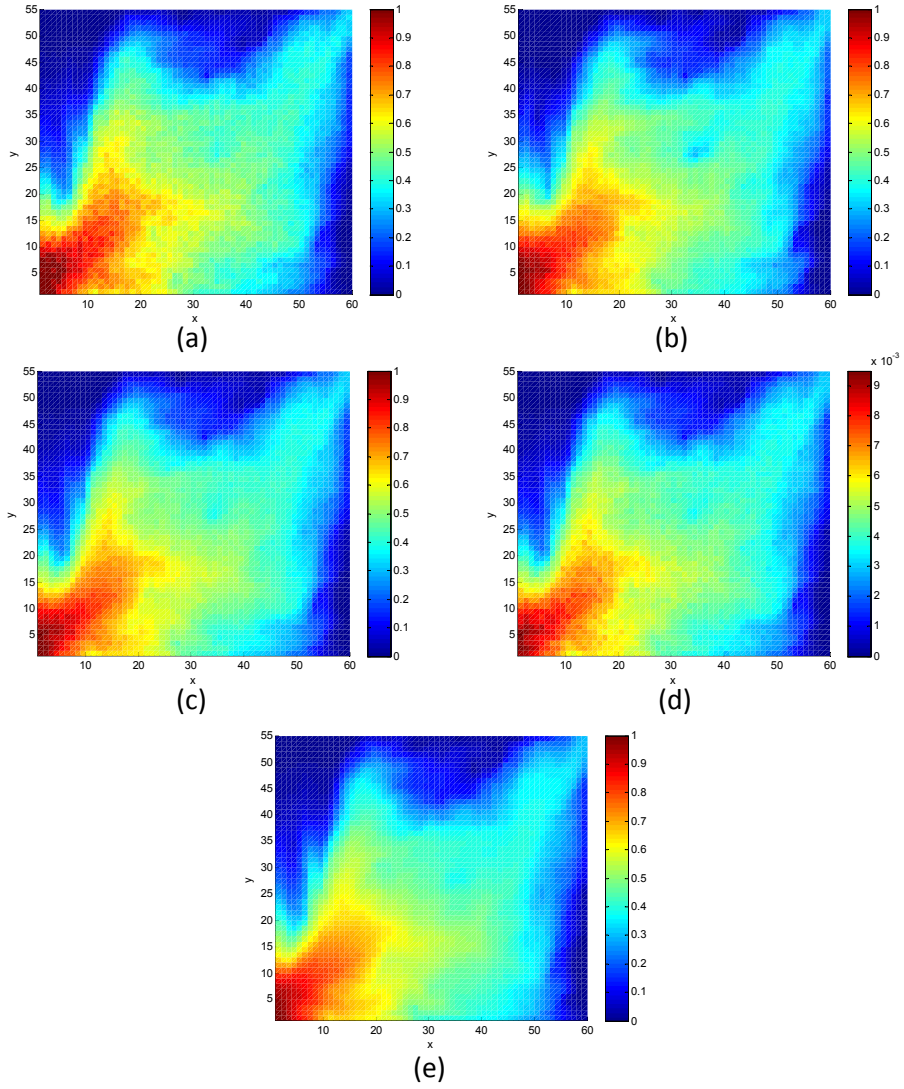

Figure 4.19: Porous media flow - comparison of mean predictions of the mean of the saturation at $T = 2000$ days provided by the model with (a) $n_\xi = 40$; (b) $n_\xi = 80$; (c) $n_\xi = 160$, to the MC result with (e) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the mean of the saturation at $T = 2000$ days for $n_\xi = 160$ observations.

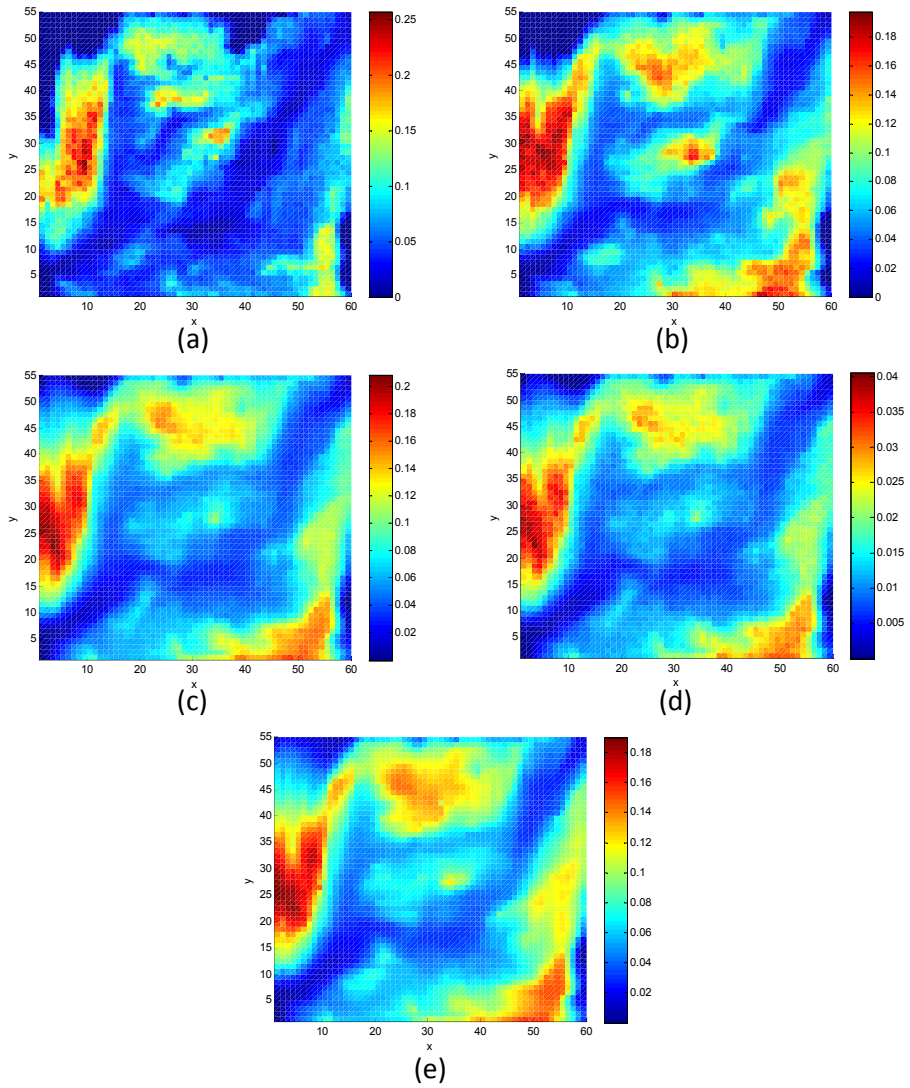

Figure 4.20: Porous media flow - comparison of mean predictions of the std of the saturation at $T = 2000$ days provided by the model with (a) $n_\xi = 40$; (b) $n_\xi = 80$; (c) $n_\xi = 160$, to the MC result with (e) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the std of the saturation at $T = 2000$ days for $n_\xi = 160$ observations.

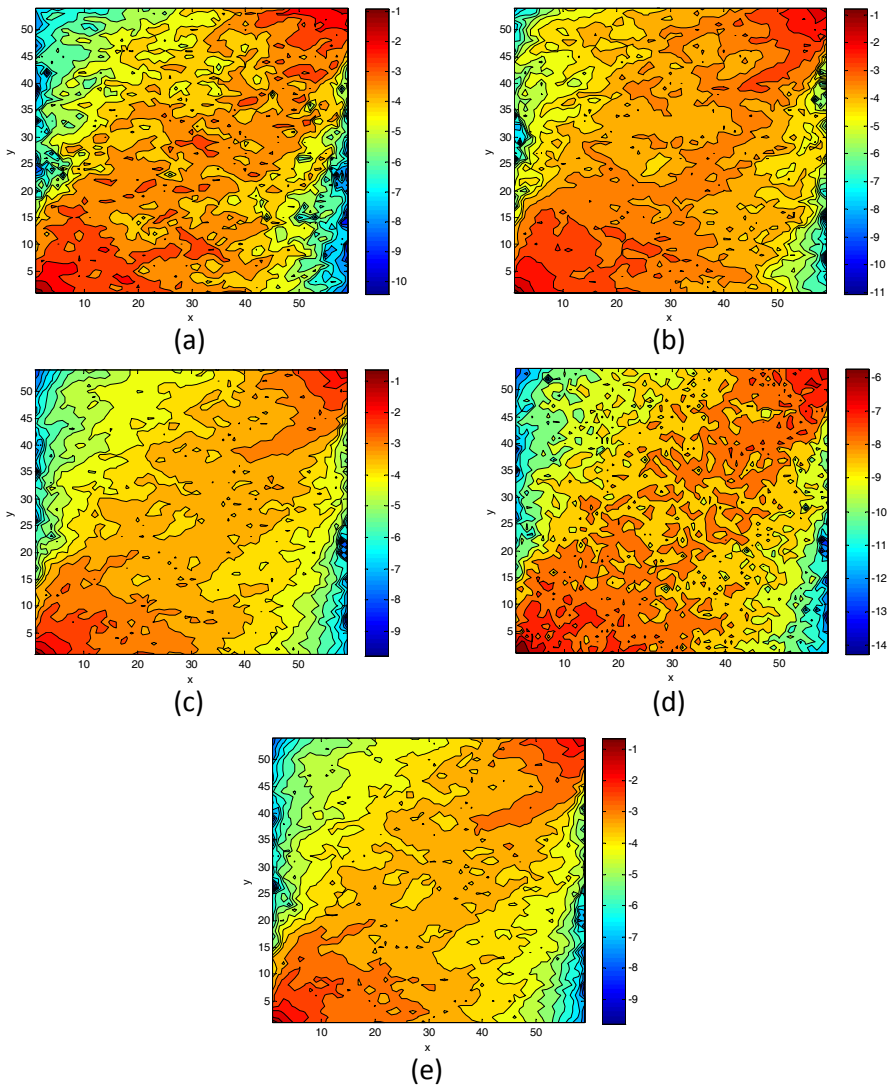

Figure 4.21: Porous media flow - comparison of mean of the mean of the natural log of the x-velocity component at $T = 2000$ days provided by the model with (a) $n_\xi = 40$; (b) $n_\xi = 80$; (c) $n_\xi = 160$, to the MC result with (e) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the mean of the natural log of x-velocity component at $T = 2000$ days for $n_\xi = 160$ observations.

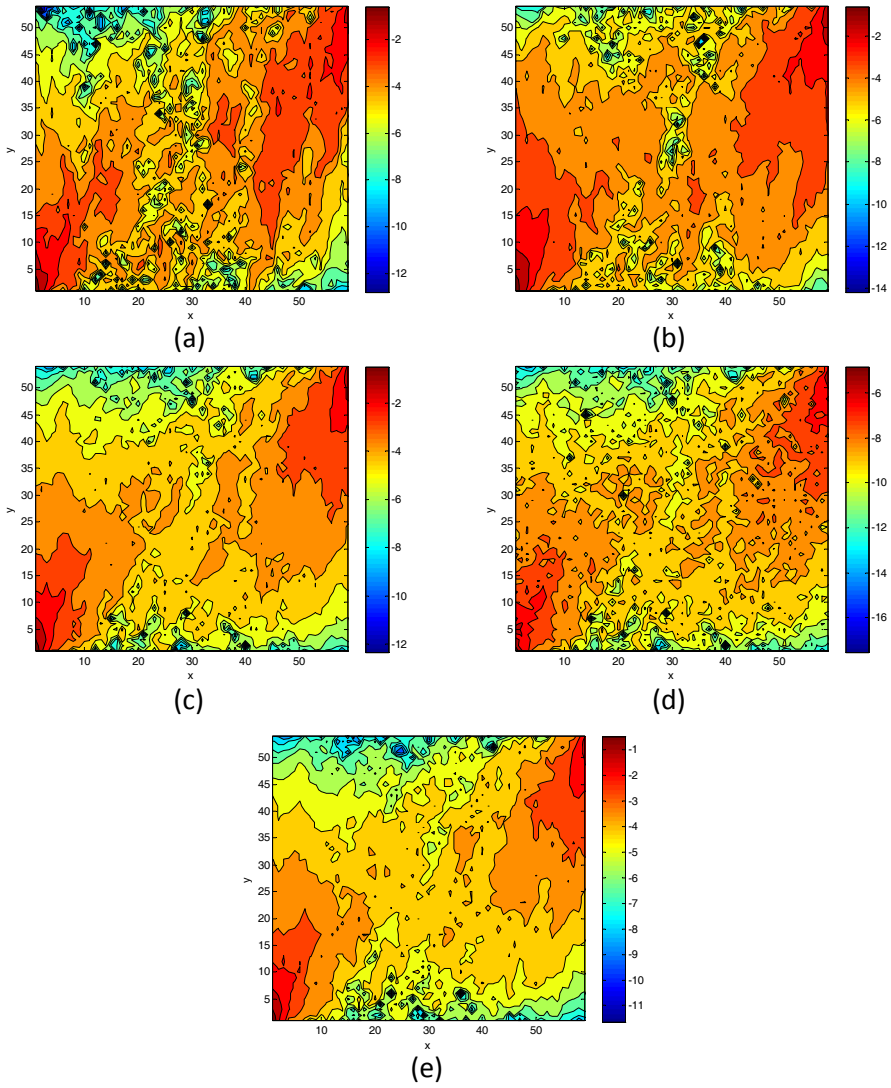

Figure 4.22: Porous media flow - comparison of mean predictions of the std of the natural log of the x-velocity component at $T = 2000$ days with (a) $n_\xi = 40$; (b) $n_\xi = 80$; (c) $n_\xi = 160$, to the MC result with (e) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the std of the natural log of x-velocity component at $T = 2000$ days for $n_\xi = 160$ observations.

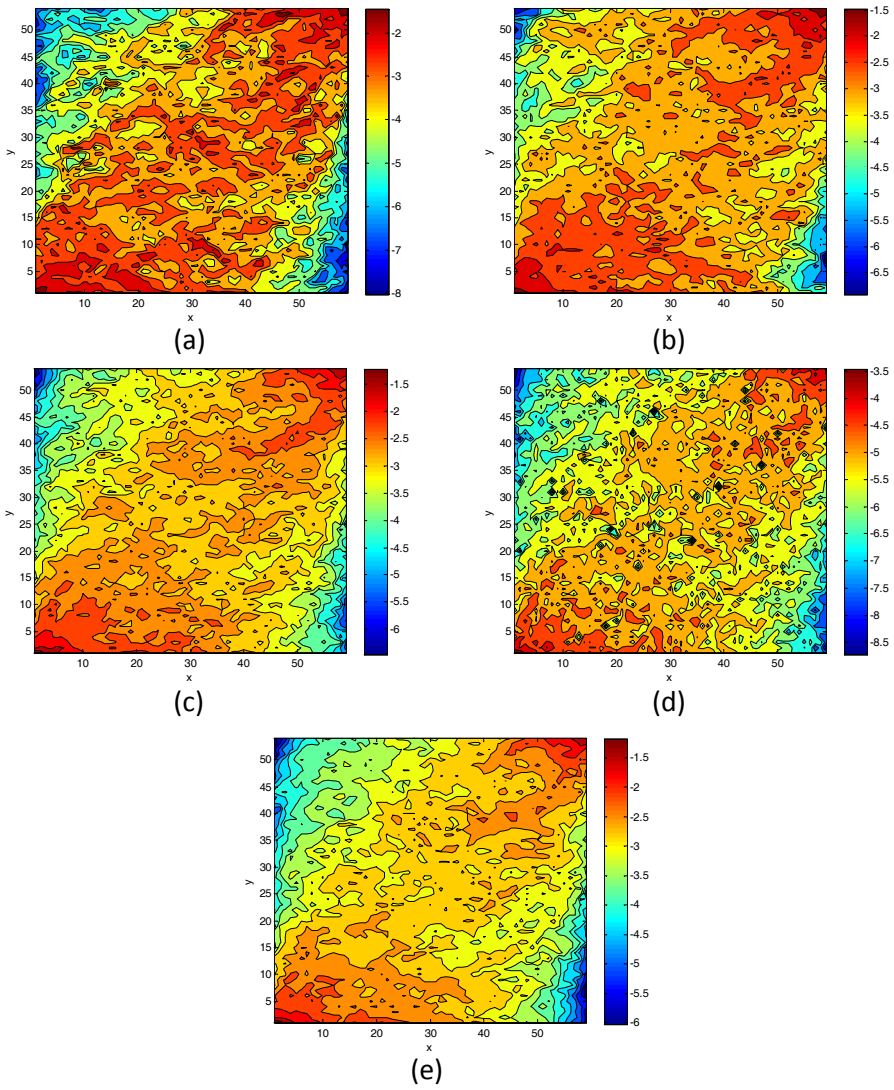

Figure 4.23: Porous media flow - comparison of mean of the mean of the natural log of the y-velocity at $T = 2000$ days provided by the model with (a) $n_\xi = 40$; (b) $n_\xi = 80$; (c) $n_\xi = 160$, to the MC result with (e) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the mean of the natural log of the y-velocity at $T = 2000$ days for $n_\xi = 160$ observations.

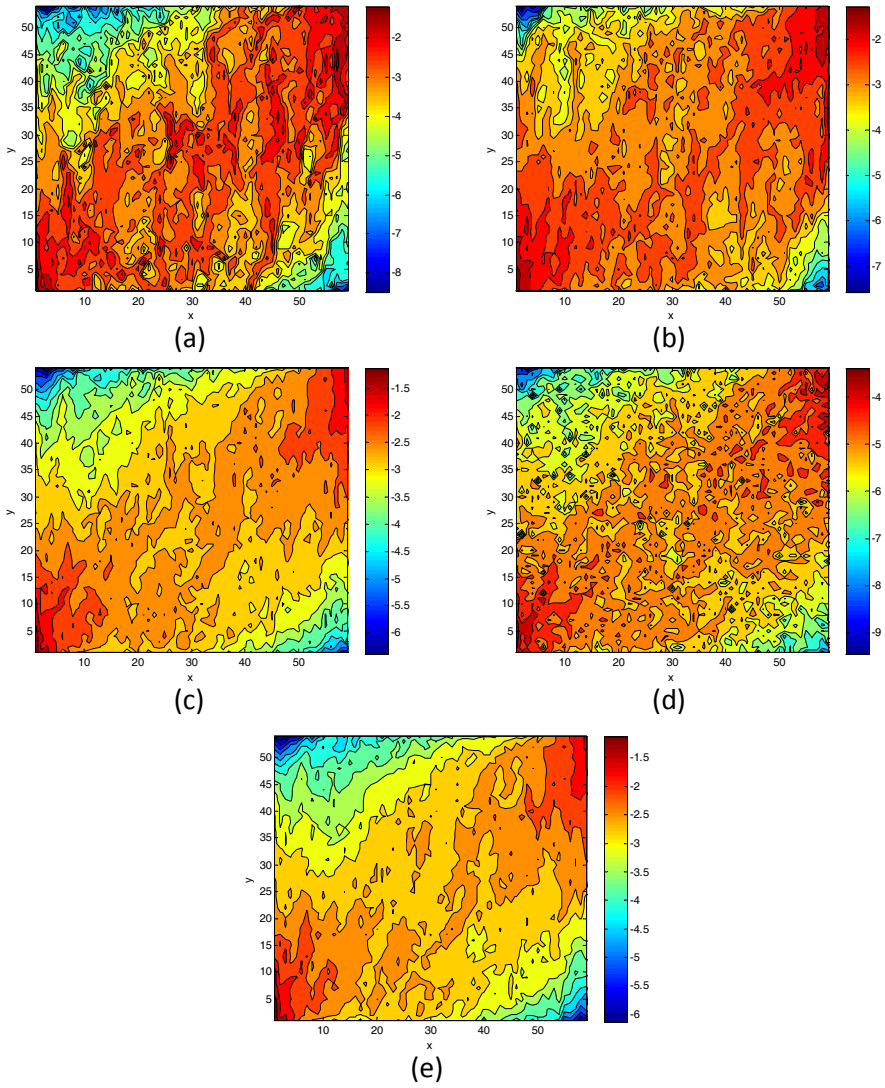

Figure 4.24: Porous media flow - comparison of mean predictions of the std of the natural log of the y-velocity at $T = 2000$ days provided by the model with (a) $n_\xi = 40$; (b) $n_\xi = 80$; (c) $n_\xi = 160$, to the MC result with (e) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the std of the natural log of the y-velocity component at $T = 2000$ days for $n_\xi = 160$ observations.

The calculation of the predicted probability densities is the same as for the KO problem. Figs. 4.25 and 4.26 show the predicted PDFs of the saturation at $T = 1000$ and $T = 2000$ days, respectively, at various spatial locations with dif-

ferent number of training data, and compare them to the MC estimates with 10^5 observations. Subfigures (a) show the probability densities at location $(10, 10)$ on the spatial grid. We can observe that the two distinct tails of the distribution are gradually captured by increasing the number of observations. This demonstrates that the proposed framework has a better performance in the prediction of PDFs compared to the results in [11]. Subfigure (b) plots the densities in the middle of the mesh. Note that there are no negative saturation values in the samples. A small peak around zero in Fig. 4.25(b) is simply given by the kernel density estimator which tends to provide a smooth representation. One can observe that with only $n_\xi = 160$ observations, the PDFs can be accurately captured.

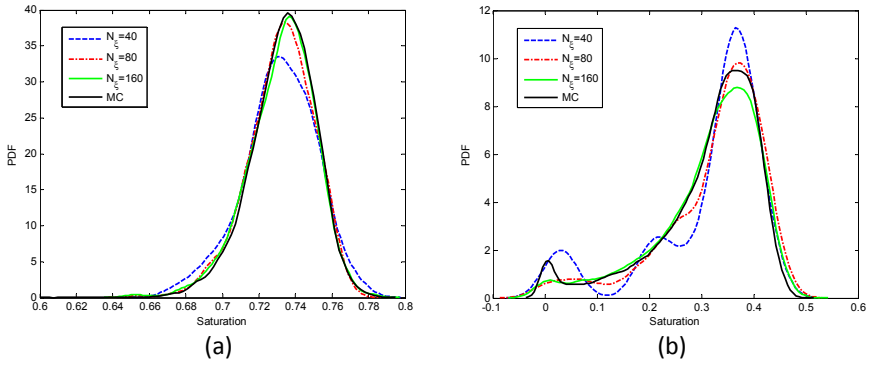

Figure 4.25: Porous media flow - comparison of mean predictions of the PDFs of the saturation at various locations at $T = 1000$ days provided by the model to the MC results, (a) at location $(10, 10)$; (b) at location $(30, 22)$.

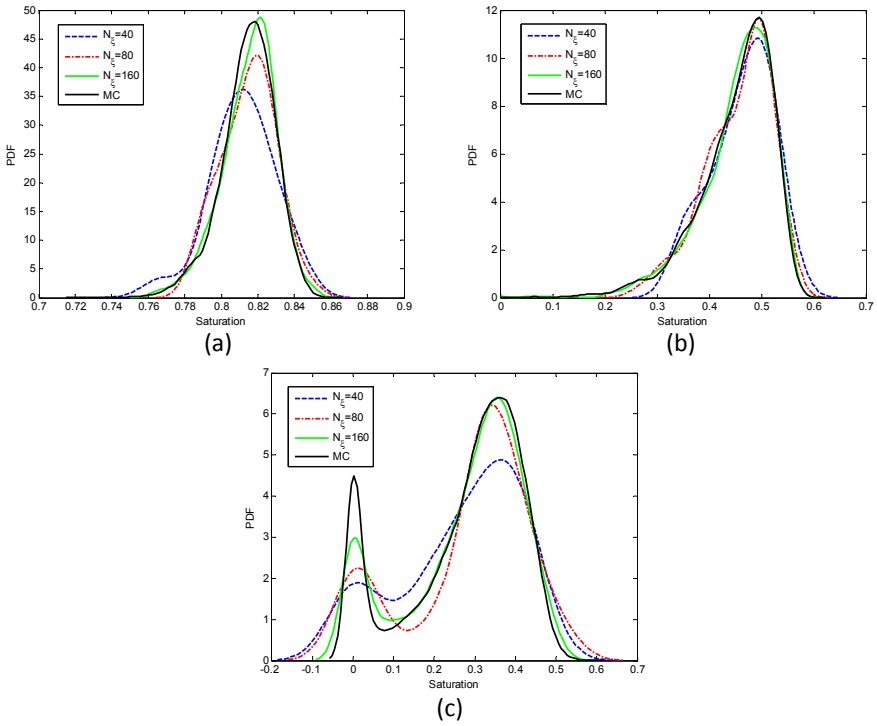

Figure 4.26: Porous media flow - comparison of mean predictions of the PDFs of the saturation at various locations at $T = 2000$ days provided by the model to the MC results, (a) at location $(10, 10)$; (b) at location $(30, 22)$; (c) at location $(5, 50)$.

The water-cut curve represents how much oil is produced at each time in the form of the fractional flow $F(t)$ defined in Eq. (4.103). The prediction of the water-cut curve is calculated from the predicted velocity and saturation. The process of obtaining the predictions is the same as above. For each sampled surrogate, we calculate the mean and variance of the water-cut curve. This step is time consuming since the velocity and saturation at every $t = 20$ days needs to be predicted for the fractional flow calculation. Fig. 4.27(a) provides a comparison of the mean prediction of the mean water-cut curve with various number of observations to the MC estimates with 10^5 observations. The comparison with MC of the standard deviation of the water-cut curve is given in Fig. 4.27(b).

From these two figures, we can conclude that the mean of the water-cut curve can be easily captured, whereas the variance can be gradually captured as we increase the number of observations.

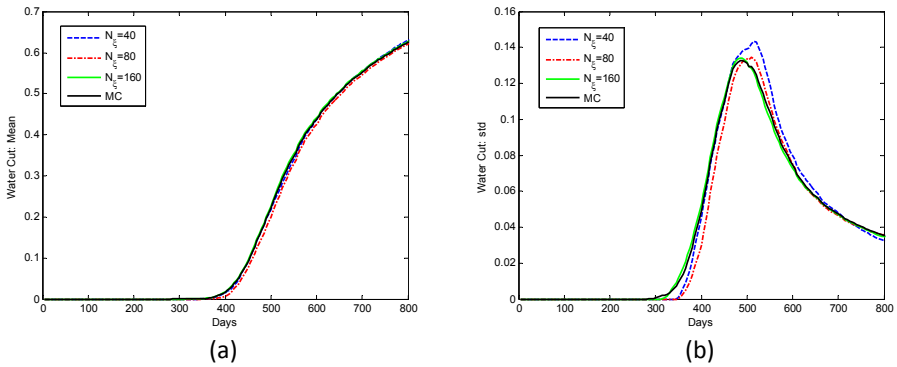

Figure 4.27: Porous media flow - comparison of the predictions of the water cut curve with different number of observations to the MC results, (a) mean predictions of mean water cut and MC estimate; (b) mean predictions of std water cut and MC estimate.

4.3 Discussion and Conclusions

A Bayesian framework based on an infinite mixture of multi-output Gaussian processes was developed to address uncertainty propagation in various problems governed by differential equations. In the flow in random media problem, the input uncertainty was assumed to come from the subsurface permeability and porosity. The outputs of interest were the flow and pressure responses. The input variables considered in the proposed framework involved not only the stochastic variables, but also the spatial and time variables. The framework had the ability to capture the non-Gaussian or local features due to the nature of the mixture model. The optimal number of mixture models could be automatically found by assigning a Dirichlet process prior. Each mixture component was

one multi-output Gaussian process model which explained the local nonlinear relationship between the inputs and responses. The posterior distribution of interest was approximated by a variational inference algorithm. A probabilistic surrogate model was then constructed to give predictions for the statistics of interest.

Various examples were considered to study the accuracy and efficiency of the proposed framework. It was shown that this framework was capable of providing reliable predictions for the statistics with rather limited number of observations. In the provided examples, it was demonstrated that this framework could correctly provide the mean predictions of the first- and second-order statistics and reasonable error bars as well. The non-Gaussian feature of the PDFs was correctly captured. It was also shown that the evolution of the uncertainty propagation over time could be efficiently predicted. Various tasks remain to be further considered including (i) approaches that narrow the predicted error bars; (ii) exploring efficient decompositions of the covariance function for each GP in the mixture; (iii) more efficient treatment of the normalization constant in the calculation of responsibilities; and (iv) integrating this work with inverse problems solving (e.g. use limited output data to predict unobservable input information).

CHAPTER 5

CONCLUSION AND SUGGESTIONS FOR FUTURE RESEARCH

In this thesis, we dealt with three important problems relevant to generic uncertainty quantification of complex physical systems - (i) how to resolve the discontinuity or local features in the stochastic space, (ii) how to address the problem of curse of dimensionality and (iii) how to extract the uncertainty information from limited number of observations. To resolve all these issues, three distinct UP frameworks were developed and demonstrated with application on physical problem governed by SPDEs. The achievements of this thesis can be summarized as: (1) constructed an efficient local framework (ALWPR) to accurately capture the local features by an actively learning scheme and quantify the uncertainty propagated to the response regardless of the form of the input uncertainty; (2) introduced a nonparametric probabilistic graphical model framework that addresses the UP problem by mapping the stochastic physical problem onto a well designed graph. A localized model reduction approach was integrated into the graphical model to extract the major uncertainty information and reduce the dimensionality of the input; (3) constructed a fully Bayesian infinite mixture of MGP model using Dirichlet process priors, which has the ability to quantify the output uncertainty, learn local discontinuities or local features, and capture the correlations between responses, with limited number of observations.

Although the three UP frameworks developed in this thesis work well for the numerical examples examined, there are still several areas where further developments and research are required. Suggestions for the continuation of this study are provided next.

5.1 Hierarchical Bayesian inference for inverse problem

The three frameworks presented above were all designed for the forward problem. With some modifications, we believe they can be extended to solve the inverse problem as well.

In general, a forward physical problem can be modeled by a nonlinear function as $\mathbf{f} : \mathcal{X} \to \mathbb{R}^q$, where $\mathcal{X} \subset \mathbb{R}^d$ is the input space. Let use $\mathbf{x} \in \mathcal{X}$ to denote the input random variables, $\mathbf{f}(\mathbf{x})$ be the prediction of the model about the physical phenomenon, and $\mathbf{y} \in \mathbb{R}^q$ to denote the corresponding experimental observation. Typically, $\mathbf{y}$ will differ from the theoretical prediction due to a variety of factors, e.g, measurement noise, model errors and etc, but most of the time, only the measurement noise will be considered [12]. The *likelihood function* is assumed to depend only on the forward solver at $\mathbf{x}$, i.e.,

$$p(\mathbf{y}|\mathbf{x}, \mathbf{f}(\cdot)) = p(\mathbf{y}|\mathbf{x}, \mathbf{f}(\mathbf{x})). \tag{5.1}$$

With the observation $\mathbf{y}$, our state of knowledge about the input $\mathbf{x}$ can be updated simply by Bayes rule:

$$p(\mathbf{x}|\mathbf{y}, \mathbf{f}(\cdot)) \propto p(\mathbf{y}|\mathbf{x}, \mathbf{f}(\mathbf{x}))p(\mathbf{x}). \tag{5.2}$$

This is the *posterior distribution* and it is the formal solution to the inverse problem. The objective is to design an efficient framework to solve the Bayesian inverse problem based on a finite set of observations. This can be achieved by replacing the forward solved with a Bayesian surrogate.

Assume we have N number of random observations

$$\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}. \tag{5.3}$$

We can make use the information contained in $\mathcal{D}$ to construct a Bayesian

surrogate model for the forward solver $\mathbf{f}(\cdot)$. Let $\mathbf{z}$ represents a q-dimensional random variable corresponding to the output of the forward model of an input $\mathbf{x}$. Then, a general Bayesian *predictive* distribution for $\mathbf{z}$ conditional on $\mathbf{x}$ and $\mathcal{D}$ is:

$$\mathbf{z}|\mathbf{x}, \mathcal{D} \sim \int \boldsymbol{p(\mathbf{z}|\mathbf{x}, \theta, \mathcal{D})p(\theta|\mathcal{D})d\theta}, \tag{5.4}$$

where θ are the hyper-parameters and $p(\mathbf{z}|\mathbf{x}, \theta, \mathcal{D})$ is the predictive distribution. $p(\theta|\mathcal{D})$ is the posterior distribution of θ given $\mathcal{D}$.

The solution of the inverse problem follows the Bayes' rule:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, \mathcal{D}) &= \int p(\mathbf{y}|\mathbf{x}, \mathbf{z})p(\mathbf{z}|\mathbf{x}, \mathcal{D})d(\mathbf{z}) \\ &\propto \int p(\mathbf{y}|\mathbf{x}, \mathbf{z})\left(\int p(\mathbf{z}|\mathbf{x}, \theta, \mathcal{D})p(\theta|\mathcal{D})d\theta\right)d(\mathbf{x}). \end{aligned} \tag{5.5}$$

This distribution, encodes all information about the input $\mathbf{x}$ w.r.t the experimental measurements $\mathbf{y}$ based on the observations $\mathcal{D}$. The objective is to develop efficient frameworks to find this distribution. Typically, there are two possible ways of achieving our goal: (1) through sampling based techniques [12]. An estimation of the target distribution could be obtained by directly sampling from it and analyzing those sample. Potential candidates for the sampling strategies are Gibbs sampling or MCMC. (2) through variational inference [14]. Variational inference algorithm has been shown in Chapter 4 to be an efficient approximation methodology for the target distribution. It over-beats both the speed and accuracy of sampling-base methods. The Bayesian surrogate model $p(\mathbf{z}|\mathbf{x}, \mathcal{D})$ could be built using any of the above three frameworks discussed in this thesis.

5.2 Multi-orthogonal model reduction

In our studies of UP problems on physical problems of interest, the biggest headache is the curse of dimensionality. For most of physical problems, the high-dimensional representation of the random field is believed to have a way to be mapped to a lower dimensional manifold, linearly or nonlinearly. Over the past few decades, a large number of dimensionaltiy reduction techniques have been proposed, see [104] for a review of current developments on dimensionality reduction methods. Nonlinear techniques exceed the linear techniques on certain aspects and can be in general divided into three groups: (1) techniques that attempt to preserve global properties of the original information, such as Isomap [99], BOD [113] and kernel PCA [65]; (2) techniques that attempt to preserve local properties of the original data, such as LLE [82], Laplacian Eigenmaps [8] and Hessian LLE [28]; (3) techniques that perform global alignment of a mixture of linear models. Among all of these, to the best of our knowledge, the bi-orthogonal decomposition (BOD) method developed by [113] is a promissing model reduction technique, especially for multiscale data. It decomposes the random field by two sets of orthogonal basis, one explains the stochastic correlations, and the other explains the spatial correlations. Potential improvement of the BOD approach is to consider a further decomposition of the random field, explicitly or implicitly, to a multiple set of orthogonal basis, with each of them explaining only partial correlations of the random field. This extension can capture the major correlations with an optimal number of random variables.

5.3 Study of UP problems with incomplete observations

In all of the above studies, we are constructing the UP framework based on fully observed data, so all the parameters, even the hidden ones, can be evaluated with nicely derived mathematical equations. But for several problems, the observation data may not be complete due to a number of reasons, e.g., incapability of measuring, highly cost of measuring and etc. Suppose $\mathbf{X} = (x_1, x_2, ..., x_d)$ denotes all the variables of interest, $\mathbf{X}^{(i)}$ denotes the i-th set of observation, here we give an example of incomplete set of observations given as follows: for each $\mathbf{X}^{(i)}$, a random element $x_j^{(i)}$ is missing for $j = 1, ..., d$. For such cases, we cannot even write the likelihood function explicitly, let alone a complete description of the UP problem. This is going to be an extremely challenging but very interesting problem. The probabilistic graphical model has the ability to solve such problems. The potential procedure is described as follows: (1) understand the meaning of all the affiliated variables and design a reasonable prior probabilistic graphical model; (2) based on the incomplete data, update the structure of the graph and compute the hyper-parameters recursively (like in the EM algorithm) until certain convergence is achieved; (3) infer the missing observations based on the optimal graphical model at convergence to complete the observation data set; (4) solve the UP problem on the graph using the complete data set as what we did in Chapter 3.

APPENDIX A

APPENDIX OF CHAPTER 2

A.1 Update of the distance metric

In Section 2.1.1, the penalized cross-validation cost function J_s (Eq. (2.13)) is defined as the leave-one-out cross-validation error J augmented with a penalty term. We can write the first term J in Eq. (2.13) as:

$$J = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i \|y_i - \tilde{y}_{i,-i}\|^2 = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i e_{i,-i}^2, \tag{A.1}$$

where $\tilde{y}_{i,-i}$ denotes the prediction at the input location $\mathbf{x}_i$ of the $i-th$ data point calculated from training the model with the $i-th$ data point $(\mathbf{x}_i, y_i)$ excluded from the training set. Also, $e_{i,-i}$ denotes the leave-one-out error. In the following, we use the subscript $()_{i,-i}$ to denote the corresponding variables to the model without using the $i-th$ data point.

In the weighted linear regression system, the parameter β can be calculated by:

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{y} = \mathbf{P}\mathbf{X}^T\mathbf{W}\mathbf{y}, \tag{A.2}$$

where $\mathbf{P}$ is the inverted weighted covariance matrix of the input. For mathematical convenience, let $\mathbf{C} = \mathbf{X}^T\mathbf{W}\mathbf{X} = \mathbf{P}^{-1}$, and $\mathbf{d} = \mathbf{X}^T\mathbf{W}\mathbf{y}$. In order to obtain $\tilde{y}_{i,-i}$, we should first compute the regression coefficient $\beta_{i,-i}$. Similarly to Eq. (A.2), we can write:

$$\boldsymbol{\beta}_{i,-i} = (\mathbf{X}_{i,-i}^T\mathbf{W}_{i,-i}\mathbf{X}_{i,-i})^{-1}\mathbf{X}_{i,-i}^T\mathbf{W}_{i,-i}\mathbf{y}_{i,-i} = \mathbf{C}_{i,-i}^{-1}\mathbf{d}_{i,-i}, \tag{A.3}$$

where

$$\begin{aligned} \mathbf{C}_{i,-i} &= \mathbf{C} - \mathbf{x}_i^T w_i \mathbf{x}_i, \\ \mathbf{d}_{i,-i} &= \mathbf{d} - w_i \mathbf{x}_i^T y_i. \end{aligned} \tag{A.4}$$

To obtain the inverse of $\mathbf{C}_{i,-i}$, we use the Sherman-Morrison-Woodbury Theorem [86]. A special case for the theorem is given below:

Sherman-Morrison-Woodbury Theorem [91] Given an invertible matrix $\mathbf{A}$ and column vector $\mathbf{v}$, then assuming $1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{v} \neq 0$,

$$(\mathbf{A} + \mathbf{v}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{v}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{v}}. \tag{A.5}$$

Using the above theorem, we can thus write:

$$\begin{aligned}
\mathbf{C}_{i,-i}^{-1} &= (\mathbf{C} - \mathbf{x}_i^T w_i \mathbf{x}_i)^{-1} = \mathbf{C}^{-1} + \frac{\mathbf{C}^{-1}\mathbf{x}_i^T w_i \mathbf{x}_i \mathbf{C}^{-1}}{1 - w_i \mathbf{x}_i \mathbf{C}^{-1}\mathbf{x}_i^T} \\
&= \mathbf{P} + \frac{\mathbf{P}\mathbf{x}_i^T w_i \mathbf{x}_i \mathbf{P}}{1 - w_i \mathbf{x}_i \mathbf{P}\mathbf{x}_i^T}.
\end{aligned} \tag{A.6}$$

Using this result and Eq. (A.3), we can express $\beta_{i,-i}$ as:

$$\begin{aligned}
\beta_{i,-i} &= \left[\mathbf{P} + \frac{\mathbf{P}\mathbf{x}_i^T w_i \mathbf{x}_i \mathbf{P}}{1 - w_i \mathbf{x}_i \mathbf{P}\mathbf{x}_i^T}\right]\left[\mathbf{d} - w_i \mathbf{x}_i^T y_i\right], \\
&= \mathbf{P}\mathbf{d} - \mathbf{P} w_i \mathbf{x}_i^T y_i + \frac{\mathbf{P}\mathbf{x}_i^T w_i \mathbf{x}_i \mathbf{P}\mathbf{d}}{1 - w_i \mathbf{x}_i \mathbf{P}\mathbf{x}_i^T} - \frac{\mathbf{P}\mathbf{x}_i^T w_i \mathbf{x}_i \mathbf{P} w_i \mathbf{x}_i^T y_i}{1 - w_i \mathbf{x}_i \mathbf{P}\mathbf{x}_i^T} \\
&= \beta + \frac{1}{1 - w_i \mathbf{x}_i \mathbf{P}\mathbf{x}_i^T}[\mathbf{P} w_i \mathbf{x}_i^T y_i w_i \mathbf{x}_i \mathbf{P}\mathbf{x}_i^T \\
&\quad -\mathbf{P} w_i \mathbf{x}_i^T y_i + \mathbf{P}\mathbf{x}_i^T w_i \mathbf{x}_i \mathbf{P}\mathbf{d} - \mathbf{P}\mathbf{x}_i^T w_i \mathbf{x}_i \mathbf{P} w_i \mathbf{x}_i^T y_i] \\
&= \beta - \frac{\mathbf{P} w_i \mathbf{x}_i^T}{1 - w_i \mathbf{x}_i \mathbf{P}\mathbf{x}_i^T}(y_i - \mathbf{x}_i \beta).
\end{aligned} \tag{A.7}$$

Here, $\mathbf{x}_i\beta = \tilde{y}_i$ is the prediction of the linear model. By multiplying the above equation by $\mathbf{x}_i$ and substracting by y_i, we obtain,

$$\begin{aligned}
e_{i,-i} = y_i - \mathbf{x}_i \beta_{i,-i} &= y_i - \mathbf{x}_i \beta + \frac{\mathbf{x}_i \mathbf{P} w_i \mathbf{x}_i^T}{1 - w_i \mathbf{x}_i \mathbf{P}\mathbf{x}_i^T}(y_i - \mathbf{x}_i \beta) \\
&= \frac{1}{1 - w_i \mathbf{x}_i \mathbf{P}\mathbf{x}_i^T}(y_i - \tilde{y}_i).
\end{aligned} \tag{A.8}$$

Hence, we can write the following:

$$J = \frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i \|y_i - \tilde{y}_{i,-i}\|^2 = \frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i \frac{\|y_i - \tilde{y}_i\|^2}{(1 - w_i \mathbf{x}_i \mathbf{P}\mathbf{x}_i^T)^2}. \tag{A.9}$$

In ALWPR, the above cost function can be formulated in terms of the PLS projected inputs $\mathbf{z}_i$ as

$$J = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i \frac{\|y_i - \tilde{y}_i\|^2}{(1 - w_i \mathbf{z}_i \mathbf{P}_z \mathbf{z}_i^T)^2}, \tag{A.10}$$

where $\mathbf{P}_z$ corresponds to the inverse covariance matrix computed from the projected inputs $\mathbf{z}_i$. The proof of $\mathbf{x}_i \mathbf{P} \mathbf{x}_i = \mathbf{z}_i \mathbf{P} \mathbf{z}_i^T$ can be found in Appendix A in [83].

A.2 Combined Prediction Variance

In LWPR, for each individual local model, we assume that the local prediction is a noisy observation of the true response with two independent noise processes [83]:

$$\tilde{y}^{(s)}(\mathbf{x}_q) = y(\mathbf{x}_q) + \epsilon_1 + \epsilon_{2,s}, \tag{A.11}$$

where $\epsilon_1 \sim \mathcal{N}(0, \sigma^2 / w_s(\mathbf{x}_q))$, $\epsilon_{2,s} \sim \mathcal{N}(0, \sigma^2_{pred,s} / w_s(\mathbf{x}_q))$ and $y(\mathbf{x}_q) = f_r(\mathbf{x}_q)$ is the true response for the output r. Recall from Eq. (2.11) that:

$$\tilde{y}(\mathbf{x}_q) = \frac{1}{\sum_s w_s(\mathbf{x}_q)} \sum_s w_s(\mathbf{x}_q) \tilde{y}^{(s)}(\mathbf{x}_q). \tag{A.12}$$

To simplify the notation, we denote in the following $w_s(\mathbf{x}_q)$ simply as w_s, and similarly $\tilde{y}(\mathbf{x}_q)$ as $\tilde{y}$, and $\tilde{y}^{(s)}(\mathbf{x}_q)$ as $\tilde{y}^{(s)}$. The combined predictive variance can now be derived as

$$\begin{aligned}
\sigma^2_{\text{pred}} &= E[\tilde{y}^2] - (E[\tilde{y}])^2 = E\left[\left(\frac{\sum_s w_s \tilde{y}^{(s)}}{\sum_s w_s}\right)^2\right] - (E[\tilde{y}])^2 \\
&= \frac{1}{(\sum_s w_s)^2} E\left[\left(\sum_s w_s y\right)^2 + \left(\sum_s w_s \epsilon_1\right)^2 + \left(\sum_s w_s \epsilon_{2,s}\right)^2\right] - (\tilde{y})^2 \\
&= \frac{1}{(\sum_s w_s)^2} E\left[\left(\sum_s w_s \epsilon_1\right)^2 + \left(\sum_s w_s \epsilon_{2,s}\right)^2\right]
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{(\sum_s w_s)^2}\left[var\left(\sum_s w_s \epsilon_1\right) + var\left(\sum_s w_s \epsilon_{2,s}\right)\right] \\
&= \frac{1}{(\sum_s w_s)^2}\left[\sum_s w_s^2 \frac{\sigma^2}{w_s} + \sum_s w_s^2 \frac{\sigma^2_{pred,s}}{w_s}\right] \\
&= \frac{\sum_s w_s \sigma^2}{(\sum_s w_s)^2} + \frac{\sum_s w_s \sigma^2_{pred,s}}{(\sum_s w_s)^2} \\
&= \frac{\sigma^2}{\sum_s w_s} + \frac{\sum_s w_s \sigma^2_{pred,s}}{(\sum_s w_s)^2}. \qquad \text{(A.13)}
\end{aligned}
$$

APPENDIX B

APPENDIX OF CHAPTER 3

B.1 Metropolis Hastings algorithm

The Metropolis Hastings (MH) algorithm can draw samples from any probability distribution, especially, it can generate samples without knowing the normalization constant [45]. Therefore, in this work, we use MH algorithm to generate samples from the righthand side in Eq. (3.27). In the following, for mathematical convenience, we use x to denote the random variable $\mathbf{s}_{(i,j)}$, and $P(x)$ to denote the target distribution.

The detailed steps are given in Algorithm 9. The main disadvantage of the MH algorithm is that the samples generated are correlated. Even though over the long term they do correctly follow $P(x)$, a set of nearby samples will be correlated with each other and not correctly reflect the distribution. This means that if we want a set of independent samples, we have to throw away the majority of samples and only take every $n - th$ sample, for some value of n (in this work, we set $n = 5$). In addition, although the Markov chain eventually converges to the desired distribution, the initial samples may follow a very different distribution, especially if the starting point is in a region of low density. So a "burn-in" period is needed, where an initial number of samples (e.g. the first $1,000$) are thrown away.

Algorithm 9: The Metropolis Hastings Algorithm

1: Pick an arbitrary probability density $Q(x'|x_t)$, where Q is the proposal jumping distribution, which suggests a new sample value x' given a sample value x_t. Here, we choose a widely used symmetric jumping distribution – Gaussian distribution centered at x_t.

2: Start with some arbitrary point x_0 as the first sample.

3: To generate a new sample x_{t+1} given the most recent sample x_t, proceed as follows:

   1. Generate a proposed new sample value x' from the jumping distribution $Q(x'|x_t)$.

   2. Calculate the acceptance ratio as:

$$r = \frac{P(x')}{P(x_t)}. \tag{B.1}$$

   3. If $r \geq 1$, accept x' by setting $x_{t+1} = x'$.

   4. Else, accept x' with probability r. That is, pick a uniformly distributed random number $u \sim \mathcal{U}[0, 1]$, and if $u \leq r$ set $x_{t+1} = x'$, else set $x_{t+1} = x_t$.

B.2 Gaussian Mixture Reduction

Given a N-components Gaussian mixture, we want to find an effectively reduced Gaussian mixture form without losing too much information from the original Gaussian mixture. The problem can be defined as follows:

$$\widetilde{f}(x) = \sum_{i=1}^{N} \widetilde{w}_i \cdot \mathcal{N}(x; \widetilde{\mu}_i, \widetilde{\Sigma}_i) \Longrightarrow f(x) = \sum_{j=1}^{M} w_j \cdot \mathcal{N}(x; \mu_j, \Sigma_j). \tag{B.2}$$

General approaches dealing with the problem of Gaussian mixture reduc-

tion can be classified into two fields. Bottom-up approaches start with a single Gaussian function and iteratively add additional components until the original mixture density is approximated appropriately (e.g. PGMR [51]). Top-down approaches take the original Gaussian mixture density and iteratively decrease the number of mixture components, either by removing single unimportant components or by merging similar components (e.g. Salmond's algorithm [85]). In addition, these algorithms can be further divided into local and global methods. Gaussian mixture reduction via clustering (GMRC [87]) method can be classified as a top-down algorithm using global information. The interested reader can refer to [87] for the detailed algorithm.

APPENDIX C

APPENDIX OF CHAPTER 4

C.1 Derivation of the posterior of hyper-parameters

We sketch the proof of the posterior of hyper-parameters (Eq. (4.18)). We consider the m-th data subset but for mathematical convenience, the subscript m is not shown. From Eq. (4.15), we write:

$$\begin{aligned} p(\mathcal{D}|\mathbf{B},\Sigma,\theta) &= \mathcal{N}_{n\times q}\left(\mathbf{Y}|\mathbf{HB},\mathbf{A},\Sigma\right) \\ &= (2\pi)^{-\frac{nq}{2}}|\Sigma|^{-\frac{n}{2}}|\mathbf{A}|^{-\frac{q}{2}}\exp\left\{-\frac{1}{2}\mathrm{Tr}[\Sigma^{-1}(\mathbf{Y}-\mathbf{HB})^T\mathbf{A}^{-1}(\mathbf{Y}-\mathbf{HB})]\right\}. \end{aligned} \tag{C.1}$$

Using the priors for $\mathbf{B}$, Σ and θ, $\pi(\mathbf{B},\Sigma,\theta) \propto \pi(\mathbf{B},\Sigma)\pi(\theta)$ and $\pi(\mathbf{B},\Sigma) \propto |\Sigma|^{-\frac{q+1}{2}}$, the posterior of $\mathbf{B}$, Σ and θ can be written as:

$$\begin{aligned} p(\mathbf{B},\Sigma,\theta|\mathcal{D}) &\propto p(\mathcal{D}|\mathbf{B},\Sigma,\theta)\pi(\mathbf{B},\Sigma,\theta) \\ &\propto \pi(\theta)|\Sigma|^{-\frac{n+q+1}{2}}|\mathbf{A}|^{-\frac{q}{2}}\exp\left\{-\frac{1}{2}\mathrm{Tr}[\Sigma^{-1}(\mathbf{Y}-\mathbf{HB})^T\mathbf{A}^{-1}(\mathbf{Y}-\mathbf{HB})]\right\}. \end{aligned} \tag{C.2}$$

Now let us define $\Phi = (\mathbf{Y}-\mathbf{HB})^T\mathbf{A}^{-1}(\mathbf{Y}-\mathbf{HB})$. Introducing $\widehat{\mathbf{B}} = (\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{A}^{-1}\mathbf{Y}$, we can simplify Φ as follows:

$$\Phi = (\mathbf{Y}-\mathbf{H}\widehat{\mathbf{B}})^T\mathbf{A}^{-1}(\mathbf{Y}-\mathbf{H}\widehat{\mathbf{B}}) + (\mathbf{B}-\widehat{\mathbf{B}})^T(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})(\mathbf{B}-\widehat{\mathbf{B}}).$$

Substitution of this expression into Eq. (C.2) results in:

$$\begin{aligned} p(\mathbf{B},\Sigma,\theta|\mathcal{D}) \propto\ & \pi(\theta)|\Sigma|^{-\frac{n+q+1}{2}}|\mathbf{A}|^{-\frac{q}{2}}\exp\left\{-\frac{1}{2}\mathrm{Tr}[\Sigma^{-1}(\mathbf{Y}-\mathbf{H}\widehat{\mathbf{B}})^T\mathbf{A}^{-1}(\mathbf{Y}-\mathbf{H}\widehat{\mathbf{B}})]\right\} \\ & \exp\left\{-\frac{1}{2}\mathrm{Tr}[\Sigma^{-1}(\mathbf{B}-\widehat{\mathbf{B}})^T(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})(\mathbf{B}-\widehat{\mathbf{B}})]\right\}. \end{aligned}$$

From this expression, we can immediately conclude Eq. (4.28) and also show that

$$\mathbf{B}|\mathcal{D}, \mathbf{\Sigma}, \theta \sim \mathcal{N}_{p\times q}\left(\mathbf{B}|\widehat{\mathbf{B}}, (\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})^{-1}, \mathbf{\Sigma}\right) \tag{C.3}$$

where $\widehat{\mathbf{B}}$ is given by Eq. (4.26).

Integrating $\mathbf{B}$ out of Eq. (C.2) gives:

$$\begin{aligned}
p(\Sigma, \theta|\mathcal{D}) &\propto \pi(\theta)|\Sigma|^{-\frac{n+q+1}{2}}|\mathbf{A}|^{-\frac{q}{2}}\exp\left\{-\frac{1}{2}\mathrm{Tr}[\Sigma^{-1}(\mathbf{Y}-\mathbf{H}\widehat{\mathbf{B}})^T\mathbf{A}^{-1}(\mathbf{Y}-\mathbf{H}\widehat{\mathbf{B}})]\right\} \\
&\quad \int \exp\left\{-\frac{1}{2}\mathrm{Tr}[(\mathbf{B}-\widehat{\mathbf{B}})^T(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})(\mathbf{B}-\widehat{\mathbf{B}})]\right\} d\mathbf{B} \\
&\propto \pi(\theta)|\Sigma|^{-\frac{n+q+1}{2}}|\mathbf{A}|^{-\frac{q}{2}}\exp\left\{-\frac{1}{2}\mathrm{Tr}[\Sigma^{-1}(\mathbf{Y}-\mathbf{H}\widehat{\mathbf{B}})^T\mathbf{A}^{-1}(\mathbf{Y}-\mathbf{H}\widehat{\mathbf{B}})]\right\} \\
&\quad |\Sigma|^{\frac{p}{2}}|\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H}|^{-\frac{q}{2}} \\
&\propto \pi(\theta)|\Sigma|^{-\frac{n-p+q+1}{2}}|\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H}|^{-\frac{q}{2}}|\mathbf{A}|^{-\frac{q}{2}} \\
&\quad \exp\left\{-\frac{1}{2}\mathrm{Tr}[\Sigma^{-1}(\mathbf{Y}-\mathbf{H}\widehat{\mathbf{B}})^T\mathbf{A}^{-1}(\mathbf{Y}-\mathbf{H}\widehat{\mathbf{B}})]\right\}.
\end{aligned} \tag{C.4}$$

From the above expression, we can verify that the posterior of Σ follows an inverse-Whishart distribution with q dimensions and n degrees of freedom:

$$\mathbf{\Sigma}|\mathcal{D}, \theta \sim \mathcal{W}_q^{-1}((n-p)\mathbf{\Sigma}|\mathbf{W}, n-p), \tag{C.5}$$

where

$$\mathbf{W} = \frac{1}{n-p}(\mathbf{Y}-\mathbf{H}\widehat{\mathbf{B}})^T\mathbf{A}^{-1}(\mathbf{Y}-\mathbf{H}\widehat{\mathbf{B}}). \tag{C.6}$$

The posterior $p(\theta|\mathcal{D}, \mathbf{B}, \Sigma)$ can be computed as $p(\theta)p(\mathcal{D}|\mathbf{B}, \Sigma, \theta)$, where the likelihood $p(\mathcal{D}|\mathbf{B}, \Sigma, \theta)$ is computed from Eq. (4.15) and the prior $p(\theta)$ from Eq. (4.13). This is a computionally non-tractable posterior that in Section 4.1.4 is approximated by variational inference. The posteriors $p(\mathbf{B}|\mathcal{D}, \Sigma, \theta)$, $p(\Sigma|\mathcal{D}, \theta)$ and $p(\theta|\mathcal{D}, \mathbf{B}, \Sigma)$ are essential in the implementation of the variational inference algorithm of Section 4.1.4.

C.2 Variational Inference: Proof of Eq. (4.57)

We want to show that: $\ln p(\mathcal{D}) = KL[q(\Psi)||p(\Psi|\mathcal{D})] + \mathcal{L}(q, \mathcal{D})$. Following simple algebra, one can show:

$$\begin{aligned}
\ln p(\mathcal{D}) &= \ln \frac{p(\mathcal{D}, \Psi)}{p(\Psi|\mathcal{D})} \\
&= \int q(\Psi) \ln \frac{p(\mathcal{D}, \Psi)}{p(\Psi|\mathcal{D})} d\Psi \\
&= \int q(\Psi) \ln \left(\frac{p(\mathcal{D}, \Psi)}{p(\Psi|\mathcal{D})} \frac{q(\Psi)}{q(\Psi)} \right) d\Psi \\
&= \int q(\Psi) \left(\ln \frac{q(\Psi)}{p(\Psi|\mathcal{D})} + \ln \frac{p(\mathcal{D}, \Psi)}{q(\Psi)} \right) d\Psi \\
&= \int q(\Psi) \ln \frac{q(\Psi)}{p(\Psi|\mathcal{D})} d\Psi + \int q(\Psi) \ln \frac{p(\mathcal{D}, \Psi)}{q(\Psi)} d\Psi \\
&= KL[q(\Psi)||p(\Psi|\mathcal{D})] + \mathcal{L}(q, \mathcal{D}).
\end{aligned}$$

C.3 Variational Inference: Proof of Eq. (4.61)

We are interested to compute a distribution of the form $q(\Psi) = \prod_k q_k(\omega_k)$ that maximizes the lower bound $\mathcal{L}(q, \mathcal{D})$. Denoting for simplicity $q_k(\omega_k) = q_k$, we can write:

$$\begin{aligned}
\mathcal{L}(q, \mathcal{D}) &= \int \prod_k q_k \left[\ln p(\Psi, \mathcal{D}) - \sum_k \ln q_k \right] d\Psi \\
&= \int \prod_k q_k \ln p(\Psi, \mathcal{D}) \prod_k d\omega_k - \sum_k \int \prod_j q_j \ln q_k d\omega_j \\
&= \int q_j \left[\ln p(\Psi, \mathcal{D}) \prod_{k \neq j} (q_k d\omega_k) \right] d\omega_j \\
&\quad - \int q_j \ln q_j d\omega_j - \sum_{k \neq j} \int q_k \ln q_k d\omega_k \\
&= \int q_j \ln \frac{\exp \mathbb{E}_{\mathbf{F}_\psi \setminus \omega_j} [\ln p(\Psi, \mathcal{D})]}{q_j} d\omega_j - \sum_{k \neq j} \int q_k \ln q_k d\omega_k
\end{aligned}$$

$$= \quad -\mathrm{KL}(q_j \parallel \exp \mathbb{E}_{\mathbf{F}_\psi \setminus \omega_j}[\ln p(\Psi, \mathcal{D})]) - \sum_{k \neq j} \int q_k \ln q_k d\omega_k.$$

Clearly, the lower bound $\mathcal{L}(q, \mathcal{D})$ is maximized when the Kullback-Leibler distance becomes zero, which is the case for $\ln q_j(\omega_j) = \mathbb{E}_{\mathbf{F}_\Psi \setminus \omega_j}[\ln p(\Psi, \mathcal{D})]$. The normalized distribution is finally given as follows:

$$q_j^*(\omega_j) = \frac{\exp\left(\mathbb{E}_{\mathbf{F}_\psi \setminus \omega_j}[\ln p(\Psi, \mathcal{D})]\right)}{\int \exp\left(\mathbb{E}_{\mathbf{F}_\psi \setminus \omega_j}[\ln p(\Psi, \mathcal{D})]\right) d\omega_j}. \tag{C.7}$$

C.4 Variational Inference: Proof of Eq. (4.86)

We start by using the Gaussian mixture approximation for $q(\theta_m)$:

$$\mathcal{H}[q] = -\frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{l=1}^{L} \omega_l^2 \int_{\theta_m} \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2 \mathbb{I}_{d+1}) \ln q(\theta_m) d\theta_m$$

$$= \quad -\frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{l=1}^{L} \omega_l^2 \int_{\theta_m} \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2 \mathbb{I}_{d+1}) \ln \left(\frac{1}{\sum_{j=1}^{L} \omega_j^2} \sum_{j=1}^{L} \omega_j^2 \mathcal{N}(\theta_m; \mathbf{m}_j, \sigma_j^2 \mathbb{I}_{d+1}) \right) d\theta_m.$$

Since $-\ln(x)$ is concave in x, using Jensen's inequality, we can write $-\ln(\mathbb{E}[x]) \leq \mathbb{E}[-\ln x]$. Using this result, one can show the following:

$$\begin{aligned}
\mathcal{H}[q] \quad &\geq \quad -\frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{l=1}^{L} \omega_l^2 \ln \left(\int_{\theta_m} \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2 \mathbb{I}_{d+1}) \left(\sum_{j=1}^{L} \frac{\omega_j^2 \mathcal{N}(\theta_m; \mathbf{m}_j, \sigma_j^2 \mathbb{I}_{d+1})}{\sum_{j=1}^{L} \omega_j^2} \right) d\theta_m \right) \\
&= \quad -\frac{1}{\left(\sum_{l=1}^{L} \omega_l^2\right)^2} \sum_{l=1}^{L} \omega_l^2 \ln \left(\int_{\theta_m} \sum_{j=1}^{L} \omega_j^2 \mathcal{N}(\theta_m; \mathbf{m}_j, \sigma_j^2 \mathbb{I}_{d+1}) \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2 \mathbb{I}_{d+1}) d\theta_m \right) \\
&= \quad -\frac{1}{\left(\sum_{l=1}^{L} \omega_l^2\right)^2} \sum_{l=1}^{L} \omega_l^2 \ln q_l.
\end{aligned}$$

The argument in the log function above is denoted as q_l and it can be shown that it takes the following simplified form: $q_l = \sum_{j=1}^{L} \omega_j^2 q'_{lj}$, $q'_{lj} = \mathcal{N}(\mathbf{m}_l; \mathbf{m}_j, (\sigma_l^2 + \sigma_j^2)\mathbb{I}_{d+1})$. Indeed, we can prove this using the normalization of the multivariate

Gaussian and simple algebra:

$$
\begin{aligned}
q_l &= \int_{\theta_m} \sum_{j=1}^{L} \omega_j^2 \mathcal{N}(\theta_m; \mathbf{m}_j, \sigma_j^2 \mathbb{I}_{d+1}) \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2 \mathbb{I}_{d+1}) d\theta_m \\
&= \sum_{j=1}^{L} \omega_j^2 \int_{\theta_m} \frac{1}{(2\pi\sigma_j^2)^{(d+1)/2}} \exp\left(-\frac{1}{2\sigma_j^2} (\theta_m - \mathbf{m}_j)^T (\theta_m - \mathbf{m}_j) \right) \\
&\quad \frac{1}{(2\pi\sigma_l^2)^{(d+1)/2}} \exp\left(-\frac{1}{2\sigma_l^2} (\theta_m - \mathbf{m}_l)^T (\theta_m - \mathbf{m}_l) \right) d\theta_m \\
&= \sum_{j=1}^{L} \omega_j^2 \int_{\theta_m} \frac{1}{(2\pi\sigma_j\sigma_l)^{(d+1)}} \exp\left\{ -\frac{1}{2} \left[\frac{\sigma_j^2 + \sigma_l^2}{\sigma_j^2 \sigma_l^2} \theta^{(m)^T} \theta_m \right.\right. \\
&\quad \left.\left. -2\theta^{(m)^T} \left(\frac{\mathbf{m}_l}{\sigma_l^2} + \frac{\mathbf{m}_j}{\sigma_j^2} \right) + \left(\frac{\mathbf{m}_l^T \mathbf{m}_l}{\sigma_l^2} + \frac{\mathbf{m}_j^T \mathbf{m}_j}{\sigma_j^2} \right) \right] \right\} d\theta_m \\
&= \sum_{j=1}^{L} \frac{\omega_j^2}{(2\pi(\sigma_j^2 + \sigma_l^2))^{(d+1)/2}} \exp\left\{ -\frac{1}{2(\sigma_j^2 + \sigma_l^2)} \left[\mathbf{m}_l^T \mathbf{m}_l - 2\mathbf{m}_l^T \mathbf{m}_j + \mathbf{m}_j^T \mathbf{m}_j \right] \right\} \\
&= \sum_{j=1}^{L} \omega_j^2 \mathcal{N}(\mathbf{m}_l; \mathbf{m}_j, (\sigma_l^2 + \sigma_j^2) \mathbb{I}_{d+1}).
\end{aligned}
$$

C.5 Variational Inference: Proof of Eq. (4.89) (Multivariate Delta Method for Moments)

Substituting Eq. (4.88) into Eq. (4.87) results in the following:

$$
\begin{aligned}
&\mathbb{E}_{\theta_m}[g(\theta_m)] \approx \frac{1}{\sum_{l=1}^{L} \omega_j^2} \sum_{l=1}^{L} \omega_j^2 \int_{\theta_m} \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2 \mathbb{I}_{d+1}) \hat{g}_l(\theta_m) d\theta_m \\
= &\frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{l=1}^{L} \omega_j^2 \int_{\theta_m} \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2 \mathbb{I}_{d+1}) \left(g(\mathbf{m}_l) + \nabla g(\mathbf{m}_l)(\theta_m - \mathbf{m}_l) \right. \\
&\left. + \frac{1}{2} (\theta_m - \mathbf{m}_l)^T \mathcal{H}_l (\theta_m - \mathbf{m}_l) \right) d\theta_m \\
= &\frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{l=1}^{L} \omega_j^2 \left\{ g(\mathbf{m}_l) + \int_{\theta_m} \mathcal{N}(\theta_m; \mathbf{m}_l, \sigma_l^2 \mathbb{I}_{d+1}) \frac{1}{2} (\theta_m - \mathbf{m}_l)^T \mathcal{H}_l (\theta_m - \mathbf{m}_l) d\theta_m \right\}
\end{aligned}
$$

. (C.8)

Introducing $\bar{\theta} = \theta_m - \mathbf{m}_l$, we can rewrite the second term as:

$$\begin{aligned}\mathbb{E}_{\theta_m}\left[\frac{1}{2}(\theta_m - \mathbf{m}_l)^T \mathcal{H}_l (\theta_m - \mathbf{m}_l)\right] &= \mathbb{E}_{\bar{\theta}}\left[\frac{1}{2}\bar{\theta}^T \mathcal{H}_l \bar{\theta}\right] \\ &= \frac{1}{2}\sum_i \mathbb{E}_{\bar{\theta}}\left[\bar{\theta}_i \mathcal{H}_{l,ii} \bar{\theta}_i\right] + \frac{1}{2}\sum_{i\neq j} \mathbb{E}_{\bar{\theta}}\left[\bar{\theta}_i \mathcal{H}_{l,ij} \bar{\theta}_j\right] \\ &= \frac{1}{2}\sum_i \mathcal{H}_{l,ii}\mathbb{E}_{\bar{\theta}}\left[\bar{\theta}_i \bar{\theta}_i\right] + \frac{1}{2}\sum_{i\neq j} \mathcal{H}_{l,ij}\mathbb{E}_{\bar{\theta}}\left[\bar{\theta}_i \bar{\theta}_j\right]. \quad \text{(C.9)}\end{aligned}$$

Due to our representation in Eq. (4.77), we can derive that $\mathbb{E}_{\bar{\theta}}\left[\bar{\theta}_i \bar{\theta}_i\right] = \sigma_l^2$, and for $i \neq j$, $\mathbb{E}_{\bar{\theta}}\left[\bar{\theta}_i \bar{\theta}_j\right] = 0$. Substitution of these results in Eqs. (C.9) and (C.8) gives Eq. (4.89).

C.6 Variational Inference: Derivation of the Derivatives $\frac{\partial \mathcal{L}_1[q]}{\partial \mathbf{m}_k}$, $\frac{\partial \mathcal{L}_2[q]}{\partial \sigma_k}$, and $\frac{\partial \mathcal{L}_2[q]}{\partial \omega_k}$ of the Lower Bound

Starting from Eq. (4.91), we first calculate $\frac{\partial \mathcal{L}_1[q]}{\partial \mathbf{m}_l}$,

$$\frac{\partial \mathcal{L}_1[q]}{\partial \mathbf{m}_l} = \frac{1}{\sum_{k=1}^L \omega_k^2}\omega_l^2 \frac{\partial g(\mathbf{m}_l)}{\partial \mathbf{m}_l} - \frac{1}{(\sum_{k=1}^L \omega_k^2)^2}\sum_{k=1}^L \frac{\omega_k^2}{q_k}\frac{\partial q_k}{\partial \mathbf{m}_l}. \quad \text{(C.10)}$$

Using the definition in Eq. (4.84) and Eqs. (C.1) and (4.13), we can write:

$$\begin{aligned} g(\theta_m) &= \ln p(\theta_m, \mathcal{D}_m) \\ &= \ln \pi(\theta_m|\gamma) + \mathbb{E}_{\mathbf{B}_m,\Sigma_m}\left[\ln p\left(\mathcal{D}_m|\theta_m, \mathbf{B}_m, \Sigma_m\right)\right] \\ &\approx -\gamma\theta_m - \frac{1}{2}\ln|\mathbf{A}_m| - \frac{1}{2}tr\left[\left(\widehat{\Sigma}_m\right)^{-1}\left(\mathbf{Y}_m - \mathbf{H}_m\widehat{\mathbf{B}}_m\right)^T \mathbf{A}_m^{-1}\left(\mathbf{Y}_m - \mathbf{H}_m\widehat{\mathbf{B}}_m\right)\right] + \text{ const }. \end{aligned}$$

Let $\widetilde{\mathbf{Y}}_m = \mathbf{Y}_m - \mathbf{H}_m\widehat{\mathbf{B}}_m$. Taking the derivative of $g(\theta_m)$ w.r.t each component of $\theta \in \theta_m$, where $\theta_m = \{r_{m,1}, \ldots, r_{m,d}, \epsilon_m\}$, results in:

$$\left.\frac{\partial g(\theta_m)}{\partial \theta}\right|_{\mathbf{m}_k} = -\gamma_\theta - \frac{q}{2}tr\left(\mathbf{A}_m^{-1}\frac{\partial \mathbf{A}_m}{\partial \theta}\right) + \frac{1}{2}tr\left[\left(\widehat{\Sigma}_m\right)^{-1}\widetilde{\mathbf{Y}}_m^T\mathbf{A}_m^{-1}\frac{\partial \mathbf{A}_m}{\partial \theta}\mathbf{A}_m^{-1}\widetilde{\mathbf{Y}}_m\right],$$

and

$$\frac{\partial g(\theta_m)}{\partial \mathbf{m}_k} = \left(\frac{\partial g(\theta_m)}{\partial r_{m,1}}, \ldots, \frac{\partial g(\theta_m)}{\partial r_{m,d}}, \frac{\partial g(\theta_m)}{\partial \epsilon_m} \right) \Bigg|_{\mathbf{m}_k}.$$

Now, let us look at the second term of Eq. (C.10). Recall that $q_k = \sum_{j=1}^{L} \omega_j^2 q'_{kj}$ was defined in Eq. (4.86), where $q'_{kj} = \mathcal{N}(\mathbf{m}_k; \mathbf{m}_j, (\sigma_k^2 + \sigma_j^2)\mathbb{I})$,

$$\begin{aligned} \frac{\partial q_k}{\partial \mathbf{m}_l} &= \frac{\partial \sum_{j=1}^{L} \omega_j^2 q'_{kj}}{\partial \mathbf{m}_l} \\ &= \begin{cases} -\sum_{j=1}^{L} \omega_j^2 q'_{lj} \frac{\mathbf{m}_l - \mathbf{m}_j}{\sigma_l^2 + \sigma_j^2}, & l = k \\ -\omega_k^2 q'_{kl} \frac{\mathbf{m}_l - \mathbf{m}_k}{\sigma_l^2 + \sigma_k^2}, & l \neq k. \end{cases} \end{aligned}$$

This completes the calculation of all the terms needed in Eq. (C.10) to evaluate $\frac{\partial \mathcal{L}_1[q]}{\partial \mathbf{m}_k}$. Next, let us discuss how to calculate $\frac{\partial \mathcal{L}_2[q]}{\partial \sigma_k}$. Starting from Eq. (4.90), we can derive that:

$$\frac{\partial \mathcal{L}_2[q]}{\partial \sigma_k} = \frac{1}{\sum_{l=1}^{L} \omega_l^2} \omega_k^2 \sigma_k tr(\mathcal{H}_k) - \frac{1}{(\sum_{l=1}^{L} \omega_l^2)^2} \sum_{l=1}^{L} \frac{\omega_l^2}{q_l} \frac{\partial q_l}{\partial \sigma_k}. \tag{C.11}$$

For the first term, we can write:

$$tr(\mathcal{H}_k) = \left(\frac{\partial^2 g(\theta_m)}{\partial \epsilon_m^2} + \sum_{i=1}^{d} \frac{\partial^2 g(\theta_m)}{\partial r_{m,i}^2} \right) \Bigg|_{\mathbf{m}_k}.$$

Each term of $\theta \in \theta_m$ in the above equation is calculated as follows:

$$\begin{aligned} \frac{\partial^2 g(\theta_m)}{\partial \theta} \Bigg|_{\mathbf{m}_k} = \quad & -\frac{q}{2} tr\left(\mathbf{A}_m^{-1} \frac{\partial^2 \mathbf{A}_m}{\partial \theta^2} - \mathbf{A}_m^{-1} \frac{\partial \mathbf{A}_m}{\partial \theta} \mathbf{A}_m^{-1} \frac{\partial \mathbf{A}_m}{\partial \theta} \right) \\ & + \frac{1}{2} tr\left[\left(\widetilde{\Sigma}_m \right)^{-1} \left(\widetilde{\mathbf{Y}}_m^T \mathbf{A}_m^{-1} \frac{\partial^2 \mathbf{A}_m}{\partial \theta^2} \mathbf{A}_m^{-1} \widetilde{\mathbf{Y}}_m - 2 \widetilde{\mathbf{Y}}_m^T \mathbf{A}_m^{-1} \frac{\partial \mathbf{A}_m}{\partial \theta} \mathbf{A}_m^{-1} \frac{\partial \mathbf{A}_m}{\partial \theta} \mathbf{A}_m^{-1} \widetilde{\mathbf{Y}}_m \right) \right]. \end{aligned}$$

The second term in Eq. (C.11) can be obtained as:

$$\frac{\partial q_l}{\partial \sigma_k} = \frac{\partial \sum_{j=1}^{L} \omega_j^2 q'_{lj}}{\partial \sigma_k}.$$

If $l \neq k$, then

$$\begin{aligned}
\frac{\partial q_l}{\partial \sigma_k} &= \frac{\partial(\omega_l^2 q'_{lk})}{\partial \sigma_k} \\
&= \omega_l^2 \frac{\partial}{\partial \sigma_k}\left[(2\pi)^{-\frac{d+1}{2}}(\sigma_k^2+\sigma_l^2)^{-\frac{d+1}{2}}\exp\left(-\frac{(\mathbf{m}_k-\mathbf{m}_l)^T(\mathbf{m}_k-\mathbf{m}_l)}{2(\sigma_k^2+\sigma_l^2)}\right)\right] \\
&= \omega_k^2\left[(2\pi)^{-\frac{d+1}{2}}\frac{\partial(\sigma_k^2+\sigma_l^2)^{-\frac{d+1}{2}}}{\partial\sigma_k}\exp\left(-\frac{(\mathbf{m}_k-\mathbf{m}_l)^T(\mathbf{m}_k-\mathbf{m}_l)}{2(\sigma_k^2+\sigma_l^2)}\right)\right. \\
&\quad \left.+(2\pi)^{-\frac{d+1}{2}}(\sigma_k^2+\sigma_l^2)^{-\frac{d+1}{2}}\frac{\partial}{\partial\sigma_k}\exp\left(-\frac{(\mathbf{m}_k-\mathbf{m}_l)^T(\mathbf{m}_k-\mathbf{m}_l)}{2(\sigma_k^2+\sigma_l^2)}\right)\right] \\
&= \omega_k^2\left[(2\pi)^{-\frac{d+1}{2}}\left(-\frac{d+1}{2}(\sigma_k^2+\sigma_l^2)^{-\frac{d+3}{2}}2\sigma_k\right)\exp\left(-\frac{(\mathbf{m}_k-\mathbf{m}_l)^T(\mathbf{m}_k-\mathbf{m}_l)}{2(\sigma_k^2+\sigma_l^2)}\right)\right. \\
&\quad \left.+(2\pi)^{-\frac{d+1}{2}}(\sigma_k^2+\sigma_l^2)^{-\frac{d+1}{2}}\exp\left(-\frac{(\mathbf{m}_k-\mathbf{m}_l)^T(\mathbf{m}_k-\mathbf{m}_l)}{2(\sigma_k^2+\sigma_l^2)}\right)\frac{\|\mathbf{m}_k-\mathbf{m}_l\|_2^2\sigma_k}{(\sigma_k^2+\sigma_l^2)^2}\right] \\
&= \omega_k^2\left[-q'_{lk}\sigma_k\frac{d+1}{\sigma_k^2+\sigma_l^2}+q'_{lk}\sigma_l\frac{\|\mathbf{m}_k-\mathbf{m}_l\|_2^2}{(\sigma_k^2+\sigma_l^2)^2}\right] \\
&= -\omega_k^2 q'_{lk}\sigma_k\left[\frac{d+1}{\sigma_k^2+\sigma_l^2}-\frac{\|\mathbf{m}_k-\mathbf{m}_l\|_2^2}{(\sigma_k^2+\sigma_l^2)^2}\right].
\end{aligned}$$

Similarly, we can derive the $l = k$ case:

$$\begin{aligned}
\frac{\partial q_k}{\partial \sigma_k} &= \frac{\partial \sum_{j=1}^L \omega_j^2 q'_{kj}}{\partial \sigma_k} \\
&= -\sum_{j=1}^L \omega_j^2 q'_{kj}\sigma_l\left(\frac{d+1}{\sigma_k^2+\sigma_j^2}-\frac{\|\mathbf{m}_k-\mathbf{m}_j\|_2^2}{(\sigma_k^2+\sigma_j^2)^2}\right).
\end{aligned}$$

Finally, to calculate $\frac{\partial \mathcal{L}_2[q]}{\partial \omega_k}$, we proceed as follows:

$$\begin{aligned}
\frac{\partial \mathcal{L}_2[q]}{\partial \omega_k} &= \frac{2\omega_k}{\sum_{l=1}^L \omega_l^2}\left[g(\mathbf{m}_k)+\frac{\sigma_k^2}{2}tr(\mathcal{H}_k)\right] \\
&\quad -\frac{2\omega_k}{(\sum_{l=1}^L \omega_l^2)^2}\sum_{l=1}^L \omega_l^2\left[g(\mathbf{m}_l)+\frac{\sigma_l^2}{2}tr(\mathcal{H}_l)\right] \\
&\quad +\frac{4\omega_k}{(\sum_{l=1}^L \omega_l^2)^3}\sum_{l=1}^L \omega_l^2 \ln q_l - \frac{1}{(\sum_{l=1}^L \omega_l^2)^2}\sum_{l=1}^L \frac{\partial \omega_l^2 \ln q_l}{\partial \omega_k},
\end{aligned}$$

where

$$\frac{\partial \omega_l^2 \ln q_l}{\partial \omega_k} = \begin{cases} 2\omega_k \ln q_k, & l = k \\ \omega_l^2 \frac{q'_{lk}}{q_l}, & l \neq k. \end{cases}$$

BIBLIOGRAPHY

[1] J. E. Aarnes, V. Kippe, K.-A. Lie, and A. B. Rustad. *Modelling of Multiscale Structures in Flow Simulations for Petroleum Reservoirs, in: G. Hasle, K.-A. Lie, E. Quak (Eds.), Geometric Modelling, Numerical Simulation, and Optimization*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[2] J.E. Aarnes, T. Gimse, and K-A Lie. An introduction to the numerics of flow in porous media using matlab. *Geometric Modelling, Numerical Simulation, and Optimization*, pages 265–306, 2007.

[3] J.E. Aarnes, S. Krogstad, and K-A Lie. A hierarchical multiscale method for two phase flow based upon mixed finite elements and nonuniform grids. *SIAM Multiscale Model Simulation*, 5(2):337–363, 2006.

[4] H. Abdi. Partial least squares regression and projection on latent structure regression (pls regression). *Wiley Interdisciplinary Reviews: Computational Statistics*, 2:97–106, 2010.

[5] G. K. Baah, A. Podgurski, and M. J. Harrold. The Probabilistic Program Dependence Graph and Its Application to Fault Diagnosis. *IEEE Transactions on Software Engineering*, 36:528–545, 2010.

[6] I. Babuska, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Review*, 52:317–355, 2010.

[7] P. Baldi. Probabilistic Graphical Models in Computational Molecular Biology. *Journal of the Italian Association for Artificial Intelligence*, 1:8–12, 2000.

[8] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

[9] I. Bilionis and N. Zabaras. Multi-output local Gaussian process regression: Applications to uncertainty quantification. *Journal of Computational Physics*, 231:5718–5746, 2012.

[10] I. Bilionis and N. Zabaras. Multidimensional Adaptive Relevance Vector Machines for uncertainty quantification. *SIAM Journal on Scientific Computing*, 34(6):B881–B908, 2012.

[11] I. Bilionis, N. Zabaras, A. Konomi, and G. Lin. Multi-output separable Gaussian process: Towards an efficient, fully Bayesian paradigm for uncertainty quantification. *Journal of Computational Physics*, 241:212–239, 2013.

[12] Ilias Bilionis and Nicholas Zabaras. Solution of inverse problems with limited forward solver evaluations: A fully Bayesian perspective. *Inverse Problems*, 30:015004, 2014.

[13] J. A. Bilmes and C. Bartels. Graphical model architectures for speech recognition. *Signal Processing Magazine, IEEE*, 22:89–100, 2005.

[14] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[15] D. Blei and M. Jordan. Variational inference for dirichlet process mixtures. *Journal of Bayesian Analysis*, 1(1):121–144, 2006.

[16] P. Bochev and R. B. Lehoucq. On Finite Element solution of the pure Neumann problem. *SIAM Rev*, 47:50–66, 2001.

[17] G. Bouchard. Efficient bounds for softmax function and applications to approximate inference in hybrid models. *NIPS 2007 Workshop for Approximate Bayesian Inference in Continuous/Hybrid Systems, Whistler, BC, Canada*.

[18] J. F. Brendan. *Graphical Models for Machine Learning and Digital Communication*. 1998.

[19] F. Brezzi, T. J. R. Hughes, L. D. Marini, and A. Masud. Mixed discontinuous Galerkin methods for Darcy flow. *SIAM J. Sci. Comput.*, 22-23:119–145, 2005.

[20] C. J. C. Burges. *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, chapter Geometric Methods for Feature Selection and Dimensional Reduction: A Guided Tour.* Kluwer Academic Publishers, 2005.

[21] A. W. Moore C. G. Atkeson and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

[22] Y. Cao and T. A. Zang. An efficient monte carlo method for optimal control problems with uncertainty. *Computational Optimization and Application*, 26:219–230, 2003.

[23] Q. Chen, W. Kinzelbach, C. Ye, and Y. Yue. Variations of permeability and pore size distribution of porous media with pressure. *Journal of Environmental Quality*, 31(2), 2002.

[24] Z. Chen and T. Y. Hou. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Mathematics of Computation*, 72:541–576, 2002.

[25] M. A. Christie and M. J. Blunt. Tenth SPE Comparative Solution Project: a comparison of upscaling techniques. *SPE Reservoir Eng Evaluat*, 4(4):308–317, 2001.

[26] S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140:640–651, 2010.

[27] A.P. Dawid. Some matrix-variate distribution theory: notational considerations and a Bayesian application. *Biometrika*, 68(1):265–274, 1981.

[28] D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Applied Mathematics*, 100(10):55915596, 2003.

[29] N. R. Draper and R. C. van Nostrand. Ridge regression and james-stein estimation: Review and comments. *Technometrics*, 21:451–466, 1979.

[30] Y. Efendiev and T. Y. Hou. *Multiscale Finite Element Methods: Theory and Applications (Surveys and Tutorials in the Applied Mathematical Sciences, Vol. 4)*. Springer, 2009.

[31] I. Farago, A. Havasi, and Z. Zlatev. Richardson-extrapolated sequential splitting and its application. *Journal of Computational and Applied Mathematics*, 226(2):218–227, 2009.

[32] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230, 1973.

[33] J. Foo and G. E. Karniadakis. Multi-element probabilistic collocation method in high dimensions. *Journal of Computational Physics*, 229:1536–1557, 2010.

[34] I. E. Frank and J. H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35:109–135, 1993.

[35] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *IJCV*, 40(1):25–47, 2000.

[36] J. S. Hunter G. E. Box, W. G. Hunter and W. G. Hunter. *Statistics for Experimenters: Design, Innovation, and Discovery, 2nd Edition*. Wiley-Interscience, 2005.

[37] M. Galass, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, and F. Rossi. *GNU Scientific Library Reference Manual*. 2009.

[38] Samuel J. Gershman, Matthew D. Hoffman, and David M. Blei. Nonparametric Variational Inference. In *29th International Conference on Machine Learning, Edinburgh, Scotland, UK*, 2012.

[39] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Courier Dover Publications, 2003.

[40] V. Ginting, F. Pereira, and A. Rahunanthan. Rapid quantification of uncertainty in permeability and porosity of oil reservoirs for enabling predictive simulation. *Mathematics and Computers in Simulation*, 99:139–152, 2014.

[41] R. B. Gramacy and H. K. H. Lee. Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103:1119–1130, 2008.

[42] M. Grigoriu. *Stochastic Systems: Uncertainty Quantification and Propagation*. Springer Series in Reliability Engineering Reliability Engineering, 2012.

[43] S. Schaal H. Hoffman and S. Vijayakumar. Local dimensionality reduction for non-parametric regression. *Neural Process Letters*, 29:109–131, 2009.

[44] K. Hackl. *IUTAM symposium on variational concepts with applications to the mechanics of materials : Proceedings of the IUTAM Symposium on Variational Concepts with Applications to the Mechanics of Materials, Bochum, Germany, September 22-26, 2008*. Dordrecht, New York, 2010.

[45] W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57:97–109, 1970.

[46] D. M. Hawkins. On the investigation of alternative regressions by principal component analysis. *Applied Statistics*, 22:275–286, 1973.

[47] F. Hecht, O. Pironneau, J. Morice, A. L. Hyaric, and K. Ohtsuka. Freefem++ manual.

[48] P. Henning and M. Ohlberger. The heterogeneous multiscale finite ellement method for elliptic homogenization problems in perforated domains. *Numerische Mathematik*, 113(4):601–629, 2009.

[49] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. An overview of the TRILINOS project. *ACM Trans. Math. Softw.*, 31:397–423, 2005.

[50] T. Y. Hou and X. Wu. A multiscale finite element method for elliptic problems in compostie materials and porous media. *Journal of Computational Physics*, 134(1):169–189, 1997.

[51] M. Huber and U. Hanebeck. Progressive Gaussian Mixture Reduction. *In 11th International Conference on Information Fusion*, pages 1–8, 2008.

[52] M. F. Huber, T. Bailey, H. Durrant-Whyte, and U. D. Hanebeck. On entropy approximation for gaussian mixture random vectors. *In Multisensor Fusion and Integration for Intelligent Systems*, pages 181–188, 2008.

[53] A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky. Efficient multiscale sampling from products of gaussian mixtures. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schlkopf, editors, *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*. MIT Press, 2003.

[54] M. Isard. PAMPAS: Real-Valued Graphical Models for Computer Vision. *Proceedings of the 2003 IEEE computer society conference on Computer vision and pattern recognition*, pages 613–620, 2003.

[55] I. T. Jolliffe. A note on the use of principal components in regression. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 31:300–303, 1982.

[56] M. I. Jordan. Graphical models. *Statistical Science*, 19(1):140–155, 2004.

[57] M.G. Kendall, A. Stuart, K. Ord, S. Arnold, and A. O'Hagan. *Kendall's Advanced Theory of Statistics, Classical Inference and the Linear Model*. A Hodder Arnold Publication. Wiley, 1998.

[58] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[59] M. Koutsokeras, I. Pratikakis, and G. Miaoulis. A web-based 3D graphical model search engine. *In Proceedings of the Eighth International Conference on Computer Graphics and Artificial Intelligence, May 11-12, Limoges, France*, pages 79–90, 2005.

[60] P. Koutsourelakis and E. Bilionis. Scalable bayesian reduced-order models for simulating high-dimensional multiscale dynamical systems. *Multiscale Modeling and Simulation*, 9:449–485, 2011.

[61] Miguel L*á*zaro-Gredilla, Steven Van Vaerenbergh, and Neil D. Lawrence. Overlapping Mixtures of Gaussian Processes for the Data Association Problem. *Pattern Recognition*, 45(5):1386–1395, 2012.

[62] M. Loève. *Probability Theory*. Springer, Berlin, 1977.

[63] A. Logg, G. N.Wells, and J. Hake. *DOLFIN: a C++/Python Finite Element Library*. 2012.

[64] X. Ma and N. Zabaras. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *Journal of Computational Physics*, 228(8):3084–3113, 2009.

[65] X. Ma and N. Zabaras. Kernel Principal Component Analysis for Stochastic Input Model Reduction. *Journal of Computational Physics*, 230:7311–7331, 2011.

[66] X. Ma and N. Zabaras. A stochastic mixed finite element heterogeneous multiscale method for flow in porous media. *Journal of Computational Physics*, 230:4696–4722, 2011.

[67] D. Maljovec, B. Wang, A. Kupresanin, G. Johannesson, V. Pascucci, and P. Bremer. Adaptive sampling with topological scores. *Int. J. for Uncertainty Quantification*, 3:119–141, 2013.

[68] L. Mathelin and T. A. Zang. Stochastic approaches to uncertainty quantification in cfd simulations. *Numerical Algorithms*, 38:209–236, 2005.

[69] L. Mathelin, T. A. Zang, and F. Bataille. Uncertainty propagation for a turbulent compressible nozzle flow using stochastic methods. *AIAA J.*, 42(8):1669–1676, 2004.

[70] P. Ming and X. Yue. Numerical methods for multiscale elliptic problems. *Journal of Computational Physics*, 214(1):421–445, 2006.

[71] J. Mooij and H. Kappen. Sufficient conditions for convergence of Loopy Belief Propagation. In *Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 396–403, Arlington, Virginia, 2005. AUAI Press.

[72] D. Kaplan N. Rubens and M. Sugiyama. *Recommender Systems Handbook: Active Learning in Recommender Systems*. Springer, 20011.

[73] Y. Nievergelt. Total least squares: State-of-the-art regression in numerical analysis. *SIAM Review*, 36:258–264, 1994.

[74] F. Nobile, R. Tempone, and C. G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46:2309–2345, 2008.

[75] E. Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[76] S. V. Poroseva and J. Letschert. Application of evidence theory to quantify uncertainty in hurricane/typhoon track forecasts. *Meteorology and Atmospheric Physics*, 97:149–169, 2007.

[77] Carl Edward Rasmussen and Zoubin Ghahramani. Infinite Mixtures of Gaussian Process Experts. In *In Advances in Neural Information Processing Systems 14*, pages 881–888. MIT Press, 2001.

[78] CE Rasmussen and CKI Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA, 1 2006.

[79] P. Raviart and J. Thomas. *A mixed finite element method for 2-nd order elliptic problems, in: I. Galligani, E. Magenes (Eds.), Mathematical Aspects of Finite*

Element Methods, Vol. 606 of Lecture Notes in Mathematics. Springer Berlin Heidelberg, Berlin, Heidelberg, 1977.

[80] J. Ross and J. Dy. Nonparametric mixture of Gaussian processes with constraints. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 1346–1354. JMLR Workshop and Conference Proceedings, May 2013.

[81] H. Rue and L. Held. *Gaussian Markov Random Fields: Theory and Applications*, volume 104 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London, 2005.

[82] L. K. Saul S. T. Roweis. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[83] A. DSouza S. Vijayakumar and S. Schaal. Incremental online learning in high dimensions. *Neural Computation*, 17:2602–2634, 2005.

[84] H. Wold S. Wold, A. Ruhe and W. J. III Dunn. The collinearity problem in linear regression. the partial least squares approach to generalized inverses. *SIAM J Sci Stat Comput*, 5:735–743, 1984.

[85] D. J. Salmond. Mixture reduction algorithms for target tracking in clutter. *In Proceedings of SPIE*, 1305:434–445, 1990.

[86] S. Schaal and C. G. Atkeson. Constructive incremental learning from only local information. *Neural Computation*, 10:2047–2084, 1998.

[87] D. Schieferdecker and M. Huber. Gaussian Mixture Reduction via Clustering. *In 12th International Conference on Information Fusion*, pages 1536–1543, 2009.

[88] J. Schiff, E. B. Sudderth, and K. Goldberg. Nonparametric belief propagation for distributed tracking of robot networks with noisy inter-distance measurements. *IEEE International Conference on Intelligent Robots and Systems*, pages 1369 – 1376, 2009.

[89] B. Scholkopf, A. J. Smola, and K. R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[90] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[91] J. Sherman and W. J. Morrison. Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix. *Annals of Mathematical Statistics*, 20:620–624, 1949.

[92] S. A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Mathematics, Doklady*, 4:240–243, 1963.

[93] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. *Communications of the ACM*, 53:95–103, 2010.

[94] J. Sun, H. Shum, and N. Zheng. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:787–800, 2003.

[95] S. Sun and X. Xu. Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):466–475, 2011.

[96] B. F. Swindel. Geometry of ridge regression illustrated. *The American Statistician*, 35:12–15, 1987.

[97] Y. W. Teh. *Dirichlet Processes: Tutorial and Practical Course*. NIPS, 2009.

[98] Y.W. Teh and M.I. Jordan. Hierarchical Bayesian Nonparametric Models with Applications. *Bayesian Nonparametrics, Cambridge University Press*, 2010.

[99] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):269–285, 2000.

[100] R. Udawalpola and M. Berggren. Optimization of an acoustic horn with respect to efficiency and directivity. *Int. J. Numer. Methods Engrg.*, 73:1571–1606, 2008.

[101] R. Udawalpola, E. Wadbro, and M. Berggren. Optimization of a variable mouth acoustic horn. *Internat. J. Numer. Methods Engrg.*, 85:591–606, 2011.

[102] J. van de Ven, F. Ramos, and G. D. Tipaldi. An integrated probabilistic model for scan-matching, moving object detection and motion estimation. *2010 IEEE International Conference on Robotics and Automation (ICRA), 3-7 May 2010*, pages 887–894.

[103] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality Reduction: A Comparative Review. Technical report, Tilburg University Technical Report, TiCC-TR 2009-005, 2009.

[104] L.J.P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review, 2008.

[105] E. Wadbro, R. Udawalpola, and M. Berggren. Shape and topology optimization of an acoustic horn–lens combination. *J. Comput. Appl. Math.*, 234:1781–1787, 2010.

[106] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, 2008.

[107] M. J. Wainwright, T.S.Jaakkola, and A.S. Willsky. Tree-based reparameterization analysis of sum-product and its generalizations. *IEEE Transactions on Information Theory*, 49(5):1120–1146, 2003.

[108] J. Wan and N. Zabaras. A probabilistic graphical model approach to stochastic multiscale partial differential equations. *Journal of Computational Physics*, (under review), 2012.

[109] X. Wan and G. E. Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209:617–642, 2005.

[110] X. Wan and G. E. Karniadakis. Multi-element generalized polynomial chaos for arbitrary probability measures. *SIAM Journal of Scientific Computing*, 28:901–928, 2006.

[111] J. Wang, Z. Zhang, and H. Zha. Adaptive manifold learning. In Advances in Neural Information Processing Systems. *The MIT Press*, 17:1473–1480, 2005.

[112] Y. Weiss and W. T. Freeman. Correctness of belief propagation in Gaussian

graphical models of arbitrary topology. *Neural Computation*, 13:2173–2200, 2001.

[113] B. Wen and N. Zabaras. A multiscale approach for model reduction of random microstructures. *Computational Materials Science*, 63:269–285, 2012.

[114] J. L. Williams and R. A. Lau. Convergence of loopy belief propagation for data association. *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2010 Sixth International Conference* , pages 175–180, 2010.

[115] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52, 1987.

[116] T. Y. Hou X. Hu, G. Lin and P. Yan. An adaptive anova-based data-driven stochastic method for elliptic pde with random coefficients. *Technical Report, Applied and Computational Mathematics, California Institute of Technology*, 2012.

[117] D. Xiu and G. E. Karniadakis. The wiener-askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24:619–644, 2002.

[118] C. Yuan and C. Neubauer. Variational Mixture of Gaussian Process Experts. pages 1897–1904, 2008.